Injury Data Linkage Toolkit

Lawrence J. Cook, PhD and Motao (Matt) Zhu, PhD
April 2022

Council of State and Territorial Epidemiologists • 2635 Century Parkway NE, Suite 700, Atlanta, GA 30345 • 770.458.3811 • www.cste.org

TABLE OF CONTENTS

Introduction 4
Funding Acknowledgement 4
Author Acknowledgements 4
Suggested Citation 4
Purpose 4
Background 5
Probabilistic Linkage: How to Guide 6
Table of Tables 6
Table of Figures 6
Example Databases 7
Selecting Variables for Probabilistic Linkage 8
Match Weights 8
Importance of Accurately Estimating Reliability 10
Are Names Required for Probabilistic Linkage? 14
Finding Hidden Information 16
Data Cleaning 17
Summary 18
Building a Linkage Model 19
Unique Identifiers 20
Example Linkage Models 21
Blocking 23
Identifying Matched Pairs 25
Graphical Method 25
High Probability Pairs 27
Controlling the False Match Rate 28
Imputed Matched Sets 29
Evaluating Linkage Results 29
Reporting Linkage Results 30
Probabilistic vs. Deterministic Linkage 30
Summary 31
References 33
Checklist of Best Practices for Data Linkage in Injury Epidemiology 35

Getting Started 35
Data Linkage Process 35
Dissemination 39
Survey of Selected Linkage Software 40
LinkSolv 40
Match*Pro 41
R: RecordLinkage 42
R: FastLink 42
R: Reclin 43
Link King 43
Link Plus 44
References 48

INTRODUCTION

Funding Acknowledgement

This toolkit was developed with support from the Centers for Disease Control and Prevention (CDC) Cooperative Agreement Numbers NU38OT000297.

Author Acknowledgements

The authors of this toolkit are Dr. Lawrence J. Cook, PhD, and Dr. Motao (Matt) Zhu, PhD. The authors recognize the contributions of the following people to this toolkit: Mia Israel and Cailyn Lingwall of the Council of State and Territorial Epidemiologists (CSTE) supported project management for development of the toolkit. Michael Bauer and Erin Sauber-Schatz reviewed drafts and provided edits. Members of the CSTE Injury Surveillance Workgroup provided questions and feedback during technical assistance sessions that informed toolkit development.

Suggested Citation

Cook LJ, Zhu MT. *CSTE Injury Data Linkage Toolkit*. Council of State and Territorial Epidemiologists; 2022. This toolkit is available online at: https://www.cste.org/members/group.aspx?id=100174

Purpose

This toolkit was developed for the Council of State and Territorial Epidemiologists (CSTE) Injury Surveillance Workgroup, and the primary intended audience is applied injury epidemiologists. The CSTE Injury Surveillance Workgroup has convened since 2015 and currently is comprised of over 160 members from state, tribal, local, and territorial (STLT) jurisdictions as well as public health organizations and academia. Over 80 STLT jurisdictions are represented in the workgroup, which provides a space for applied epidemiologists to strategically explore and improve injury surveillance methods. Examples of workgroup products can be found on the CSTE Injury Epidemiology & Surveillance Subcommittee page and in the CSTE Injury Surveillance Toolkit. The need for data linkage specific training resources for injury surveillance arose during monthly workgroup calls and during the 2020 Injury Surveillance Workgroup Planning Meeting. This toolkit aims to address this need and includes three sections:

- the **how-to guide** covers the linkage principles and linking a variety of databases such as medical and crash data, drug overdoses, and child abuse and neglect.
- the **checklist of best practices** provides targeted action items by category throughout the linkage process, such as building partnerships, selecting variables, evaluating match pairs, analyzing linked data, and sharing results with stakeholders.
- the **survey of linkage software** highlights a selected list of available software as of December 2021, comparing their capabilities, accessibility, strengths, and weaknesses.

Background

Often researchers discover the answer to their research question can only be obtained by combining multiple databases. In injury prevention and violence research, the databases are often collected by and housed at multiple institutions and may lack a common key allowing for direct database joins. For instance, a researcher may be interested in linking the motor vehicle crash (MVC) and emergency department (ED) databases to examine the medical consequences of behavioral risk factors such as impaired driving, speeding, safety belt non-use, or alcohol and drug use. One may wish to join the ED and death certificate files to determine the outcomes of person who leaves without being seen or refuses admission. Combining the ED and poison control databases can allow a researcher to determine outcomes for callers who were referred to the ED for treatment. Each of these databases contains its own unique identifier. The MVC database may have driver license numbers and occupant numbers, the ED database may have a medical record number, a death certificate number is assigned by the vital records office, and a case number is applied to each poison control call. Unfortunately, while unique, these identifiers are usually just random strings of characters assigned to a person and cannot be joined to each other using normal database structured query language (SQL) operations. In these situations, researchers need a tool, such as probabilistic linkage, that allows one to combine databases in the absence of a common unique identifier. Probabilistic linkage is a method that compares agreements and disagreements on variables common to two databases to determine the probability that two records refer to the same person and event and should be linked. Becoming proficient in probabilistic linkage will provide you with the tools to tackle the above situation and proceed with your study.

There are many examples of how combining MVC and hospital databases has been used to support traffic safety legislation, assess at-risk groups, and evaluate programs.[1-6] Another, important application of probabilistic linkage in MVC research is defining serious injuries.[7] Detailed information regarding starting a MVC-related probabilistic program are available in publications from the Centers for Disease Control and Prevention (CDC) and that National Highway Traffic Safety Administration (NHTSA).[8-9]

Probabilistic linkage has been used successfully in several other injury prevention areas as well. Examples include opioid research,[10-12] poisoning prevention,[13] homicide and suicide prevention,[14-15] and combining trauma registries with traumatic brain injury (TBI), spinal cord injury (SCI), and other hospital-based data systems.[16-19]

PROBABILISTIC LINKAGE: HOW TO GUIDE

Table of Tables

Table 1. Example Databases 7
Table 2. Match weights for sex and SSN 10
Table 3. Impact of changing m on match weight for sex 10
Table 4. Match weight by frequency of first name 12
Table 5. Hidden information in ICD-10-CM cause codes 17
Table 6. Agree and disagree weights by adding a range to date comparisons 20
Table 7. Model for linking the MVC and ED databases 21
Table 8. Model for linking ED and death certificate databases related to an injury event 22
Table 9. Model for linking ED to Death Certificates databases within one year of ED visit 23

Table of Figures

Figure 1. Match weights for hour, county, age, and name 13
Figure 2. Sensitivity and specificity by type of name and non-name information available 15
Figure 3. Example match weight histograms 26

Example Databases

To provide an applied view of probabilistic linkage we will use example databases and linkage projects to facilitate our discussion of technical topics and give guidance on how to conduct the linkage. The four example databases are displayed in Table 1.

Table 1. Example Databases

Motor Vehicle Crash	**Emergency Department**	**Poison Control**	**Death Certificates**
Time of crash	Hospital Identifier	Case Number	Death Certificate Number
Hospital identifier if transported	Date of arrival	Date of call	Decedent's name
First Aid by: bystander, police, EMS, none	Hour of arrival	Time of call	Decedent's date of birth
Was EMS called to scene of crash Y/N	Discharge date	Caller's phone number	Decedent's sex
Hospital identifier where transported	Discharge hour	Caller's name	Decedent's address: street, city, county, zip
City/County/Zip code of crash	First and last name	Age of caller	Race/ethnicity
Type of vehicle: passenger car, motorcycle etc.	Date of birth	Relation to patient	Date and time of death
Driver's license number	Age in years	Patient's name	Location of death: city, county, state
Vehicle make/model/year	Sex	Patient's age	Injury related: yes/no
Driver contributing factors	Medical Record Number	Patient's sex	Date of injury
First and last name	ICD-10-CM Dx Codes	Substance	ICD-10 codes
Date of birth	ICD-10-CM Cause Codes	Route	Cause code if injury related
Age in years	Billing Zip Code	Intentionality of poisoning	Type of death: natural, accidental, self-harm, assault, undetermined
Sex		Address of incident	
Person type: driver, passenger, pedestrian, bicyclist		Recommended hospital treatment	

The MVC database provides information collected at the scene of the crash and contains data collected at the crash level (same for everyone involved in a crash, such as date, time, and location of crash), vehicle level (same for everyone in a vehicle, such as speed at impact and was the driver impaired), and person level (specific to each person in the crash, such as age, sex, safety restraint use, and seating position). MVC databases are typically managed by departments of transportation or public safety. The emergency department (ED) database contains data on all ED visits that did not result in hospital admission and is compiled from billing records. ED databases can typically be obtained from state health departments or hospital associations. The death certificate database is obtained from a state's office of vital records and contains information on the decedent and factors related to the death. The poison control database was obtained from a call center and contains data on the caller, the person who is affected, and recommendations provided.

Selecting Variables for Probabilistic Linkage

The success of any probabilistic linkage project depends on the size and quality of the databases as well as the accuracy and completeness of the variables used in the linkage algorithm. Thus, the most time intensive part of probabilistic linkage is usually cleaning and preparing data. Therefore, we will begin our discussion of linkage by looking at it from the variable level and gradually expand our scope to evaluating the results from our probabilistic model.

Identifying which variables will be included in your probabilistic linkage is one of the most important steps in any project. For event-based linkages it is important to include a mix of both event and person level fields. Too many event-level fields in an MVC-based linkage can result in occupants from the same vehicle getting cross-linked to each other's ED records. Similarly, too many person-level fields can result in linking a person's MVC record to every ED visit the person had that year.

Typical event-level fields include the date, time, and location of the event. These fields can be captured at a variety of specificity. For instance, time may be captured as the full clock time or just as the hour of the day. Location might be the county, city, zip code, or even the exact latitude and longitude of the event. While having more specific fields frequently seems to be more desirable, it is important to remember that the more precisely something is coded the more opportunities there are for a disagreement to occur when making comparisons between fields in a record pair. For instance, the time of an MVC is unlikely to be the same as the time of an ED visit, while it is much more likely these two events occur within the same hour. We will discuss how to include tolerances in your linkage comparisons to account for a situation where you do not expect the events to occur simultaneously. First, though, we begin with a discussion of calculating match weights as this is the metric used to judge the quality and importance of potential fields.

Match Weights

While the goal of this guide to is be largely practical, we will cover small amounts of theory when it aids the understanding of how to apply probabilistic linkage. When we asses a potential field, we care about two properties: reliability and discriminating power. Reliability, often denoted by the letter m, is the

probability that a variable agrees between a pair of records given they are a true match. Reliability can be thought of as (1 – probability of a miscode) although there are other ways a field can disagree on a true match, such as comparing county of crash to county of hospital when the patient is transported across county lines. Discriminating power, denoted by the letter u, is the probability that a field will agree between a pair of records given the pair is not a true match. Discriminating power is analogous to the probability of agreeing by chance. The value for m is typically supplied by the programmer in most software applications and is applied at the variable level as it represents a global view of the accuracy of the field. The probability of agreeing by chance, on the other hand, is dependent on the observed value, as some values are much more common, like a last name of Smith, and others are rarer, such as a last name of Zhu. Therefore, u, is determined at the value level of each field and can be roughly estimated as (the number of times that value occurs/number of records in the file). Most software applications will calculate the u probabilities for each level of every variable. As we want to work with fields that are reliable and ones where agreement by chance is rare, values of m are typically much larger than values of u.

When a field agrees or disagrees between a pair of records, an agreement or disagreement ratio can be used to measure the contribution to a match between the record pair. When a field agrees between a pair of records, we calculate the agreement ratio that is the ratio of the probability that the pair is agrees given the pair is a true match, m, to the probability of agreement given the pair is a false match, u, is calculated as (m/u). When a field disagrees between a pair of records, we calculate the disagreement ratio that is the ratio that the pair disagrees given the pair is a true match, 1 – m, to the probability the field disagrees given the pair is a false match, 1 – u, is calculated as ((1 – u)/(1 – m)). The result of this calculation is the agreement on a field between two records results in a value larger than one, while disagreements result in values that are less than one. For technical reasons, most software applications will calculate a match weight, w, by taking the logarithm base 2, $\log_2$, of the ratio from above. In this situation, agreements between pairs of records result in values of w that are positive and disagreements result in negative values of w.

To illustrate the concepts from above we will calculate match weights for a variable with few levels, sex, and a variable with many levels, Social Security Number (SSN). Here we will assume sex only has two levels, has an accuracy of 90%, m = 0.90, and is equally divided in our database, u = 0.5. We will also assume an accuracy of 0.90 for SSN. Agreement by chance for SSN can be estimated by assuming the probability of agreeing on any digit is 0.1 (ten possible values and only one correct value) and with nine digits in an SSN, we get $(0.1)^9$. The results of agreeing and disagreeing on sex and SSN are shown in Table 2.

Table 2. Match weights for sex and SSN

	Agreement			Disagreement		
	m	*u*	*w*	(1 – *m*)	(1 – *u*)	*w*
Sex	0.9	0.5	0.85	0.1	0.5	-2.32
SSN	0.9	0.1^9	29.75	0.1	1- 0.1^9	-3.32

A few things are immediately clear from Table 2. The first is that agreement on SSN gives a lot more evidence that a pair is a true match than agreement on sex, while disagreement on SSN is only slightly more evidence the pair is not a match compared to disagreement on sex. Secondly, while sex may not provide much evidence a pair is a true match, disagreement on sex provides much more evidence the pair is not a true match. Lastly, agreement on SSN almost certainly guarantees the pair will be declared a match. The high weight assigned to SSN can be both a benefit and disadvantage in probabilistic linkage and will be discussed later.

Importance of Accurately Estimating Reliability

Choosing a field's reliability, *m*, should not be taken lightly. Many software applications supply default values of *m* and if the default is much different than the true reliability of your fields your linkage results can be negatively impacted. In Table 3 we look at the impact on the match weight for different values of *m* for sex.

Table 3. Impact of changing *m* on match weight for sex

m	*u*	Agreement Weight	Disagreement Weight
0.8	0.5	0.68	-1.32
0.9	0.5	0.85	-2.32
0.99	0.5	0.99	-5.64

The top row shows the situation where sex is moderately reliable, with *m* = 0.8. The middle row is the scenario we examined in the example above with *m* = 0.9. The final row shows a situation where sex is coded nearly perfect with *m* = 0.99. In all cases, we will consider a file with an equal split between males and females, *u* = 0.5. Looking at agreement weights we can see there is about a 30% increase in match weight as *m* changes from 0.8 to 0.99. One can think of the lower value for the agreement weight in the 0.8 scenario as the result that a file with many errors is more likely to "accidentally" agree on sex, which will reduce the impact on determining whether a pair is a true match. The uncertainty with an *m* probability of 0.8 can also be seen in the disagreement weight. If sexes are likely to disagree then finding a mismatch on sex is less likely to hurt a pair from being declared a true match. If, however, *m* = 0.99

and there is a disagreement on sex then it is very unlikely this pair is a true match, which is reflected in the disagreement weight being -5.64 – more than four times higher than when m = 0.8.

Properties of Fields that Affect Match Weights

While examining the match weights for sex is instructive, it is important to get an idea for how weights behave for other types of fields. Here we look at three examples from linkages using MVC data. Our focus will be on four commonly used fields: hour of crash, county of crash, age, and first name. In our data, hour of crash can range from 0 (midnight) to 23 (11:00 pm). There are 29 counties. Ages range from 0 to 99. There are more than 30,000 unique names. Figure 1 displays the resultant agreement weights.

For ease of comparison, we have fixed the y-axis of all graphs to the same scale. Starting in Panel A, we can see the results for crash hour. Crashes between midnight and 6:00 am are less common and therefore are assigned the highest agreement weights with the max occurring at 4:00 am, with a value of 8.0. The most common hour for crashes to occur is at 5:00 pm, which has an agreement weight of 3.2. For the most part, agreement weights are similar across all hours of the day with most having a weight between 4.0 and 6.0. We next turn our attention to the county of the crash. With only 29 counties in our example, it might seem reasonable to expect similar resulting match weights as with the hour of the crash. We can see in Panel B; however, this is not the case. The lowest agreement weight is 1.0 with the maximum weight being 11. The difference is, unlike hour, where crashes are fairly spread out across the day, County 1 accounts for almost half of all crashes and counties 1 through 5 account for 80% of all crashes. For comparing the differences in weights between the hour and county, we can raise 2 to the value of the difference in agreement weights, so a crash that agrees on the rarest county (# 29) is $2^3 = 8$ times more likely to be a match compared to a crash that occurs at the rarest hour (4:00 am).

Alternatively, a crash that occurs at the most common hour, 5:00 pm, is $2^2 = 4$ times more likely to be a true match compared to one that agrees on the most common county (#1). Occupant age is displayed in Panel C. Following from the left of the graph to the right, we see that children are less common than most adults, with match weights between 7 and 8. The most common age is 16 years, which receives the lowest agreement weight of 4.5. Match weight continues to increase across the age spectrum, finally peaking at 97 years with an agreement weight of 16. This is the highest weight we have seen yet and tells us that agreeing on the rarest ages lends more evidence that the pair is a true match than agreeing on either crash hour or county. Finally, in Panel D we have the distribution of agreement weights for first name.

The most common first name in the data set is Michael, which receives a match weight of 6. Following the graph to right we see weights steadily increase until we arrive at the five thousand names which occur twice and receive an agreement weight of 18. There are more than 10,000 names which occur only once in the database and those receive a match weight of 19. Reviewing all four panels we get a sense of how names behave in a linkage. Agreement on a very common name, such as Michael, is not as informative as agreeing on most ages, several counties, or on crashes occurring shortly after midnight. Conversely, agreement on name quickly becomes much more informative than almost every level of the

other three variables we have examined. Table 4 provides example names, their frequency, and how quickly match weight increases.

Table 4. Match weight by frequency of first name

Name	Percentage of All Records	Agreement Weight
James	1.7%	5.8
Larry	0.3%	8.2
Tracy	0.1%	9.5
Virgil	0.02%	11.8

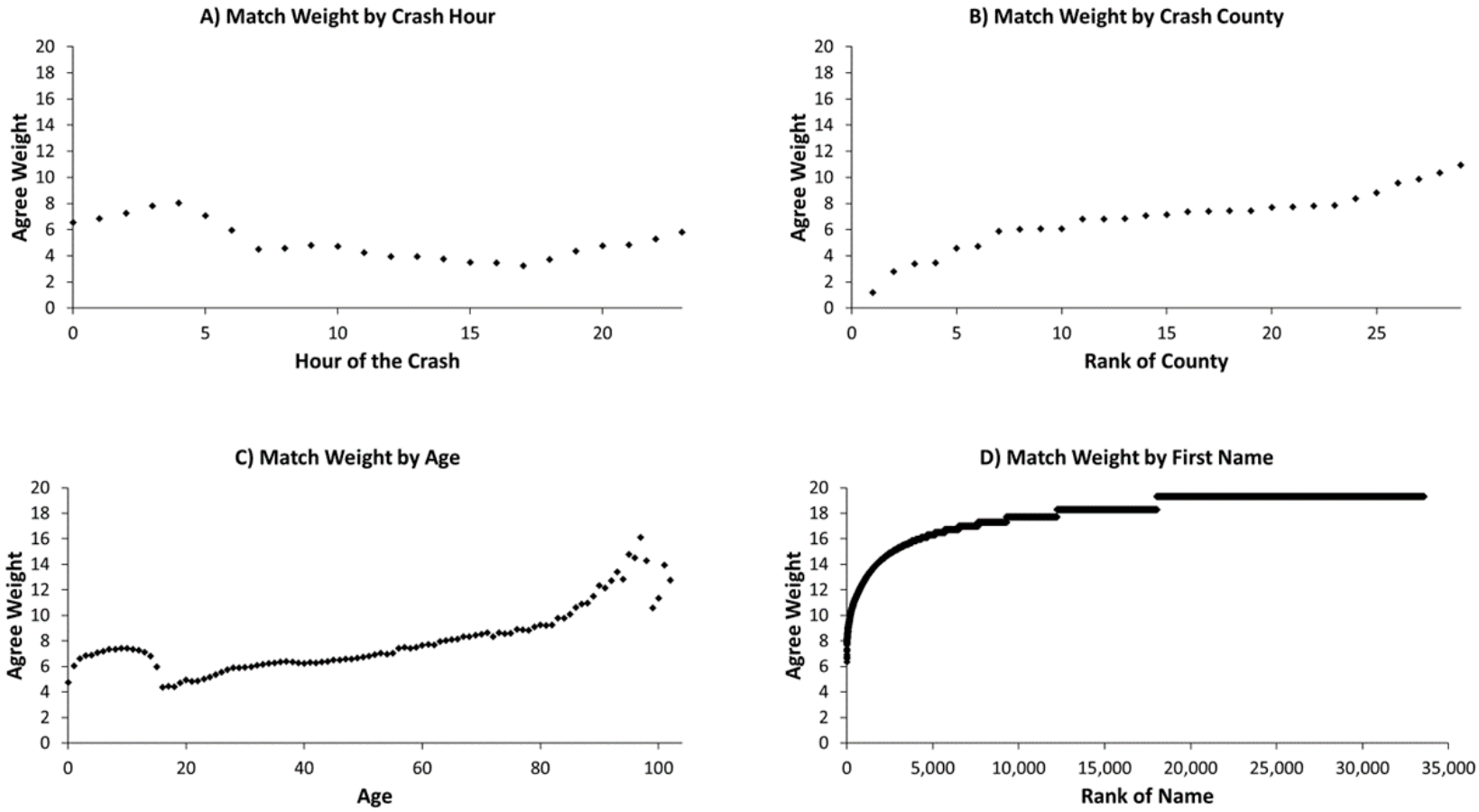

Figure 1. Match weights for hour, county, age, and name

Are Names Required for Probabilistic Linkage?

Based on the above example you may be wondering if probabilistic linkage can be successful without access to names. Here we will take a moment to describe a small experiment to determine the likelihood of a linkage succeeding without names. This study is intended to simulate linking an MVC and ED database. We generated two databases, each of 100,000 records. The records in each database were unique and did not match to any record in the other database. We then generated 10,000 more records and inserted them into each database. The result was two databases, each with 110,000 records, containing 10,000 records that exactly matched to one and only record in the other file. Fields that we included in the databases were full first and last name, date of birth and age, sex, day of crash/ED visit, time of crash/ED visit, county of crash/ED visit.

To offset the effect of using exact matches we generated errors in each of the fields. We performed linkages at several error rates: no errors (exact matches), 1%, 5%, 10%, and 25%. Errors were generated at the field level, so a 5% error rate implies five percent of names, five percent of dates, etc. have errors. Additionally, errors were required to be valid entries. In most cases, this was done by replacing the true value with a randomly generated value from that field. Because the importance of name information may depend on the other fields included in the linkage, we varied the type of non-name information available. We performed one set of linkages using date of birth, sex, and date, time, and county of crash/hospitalization. In the next set of linkages, we replaced date of birth with age; next we removed county of crash/ED visit from the linkage; and finally, we also removed the time of the crash/ED visit from the linkage. To assess the importance of names we performed linkages with four different levels of name information: full first and last name, Soundex, a phonetic algorithm for indexing names by sound which can be used to overcome common misspellings, [20] of first and last name, first and last initials, and no name information. We performed linkages with each level of name information crossed by each combination of non-name fields, and across all levels of error rates for a total of 4 x 4 x 5 = 80 linkages.

We defined two outcome measures to quantify the quality of the linkage. First, was specificity, or the ability to identify true matches, which was calculated as the (number of true matches correctly identified) / (the total number of true matches, or 10,000). Specificity is the ability to exclude false matches. Specificity was calculated as 1 – [(number of false matches) / (total number of matches found)]. A good linkage project should aim for high specificity and high sensitivity. Results of the 80 linkages are shown in Figure 2.

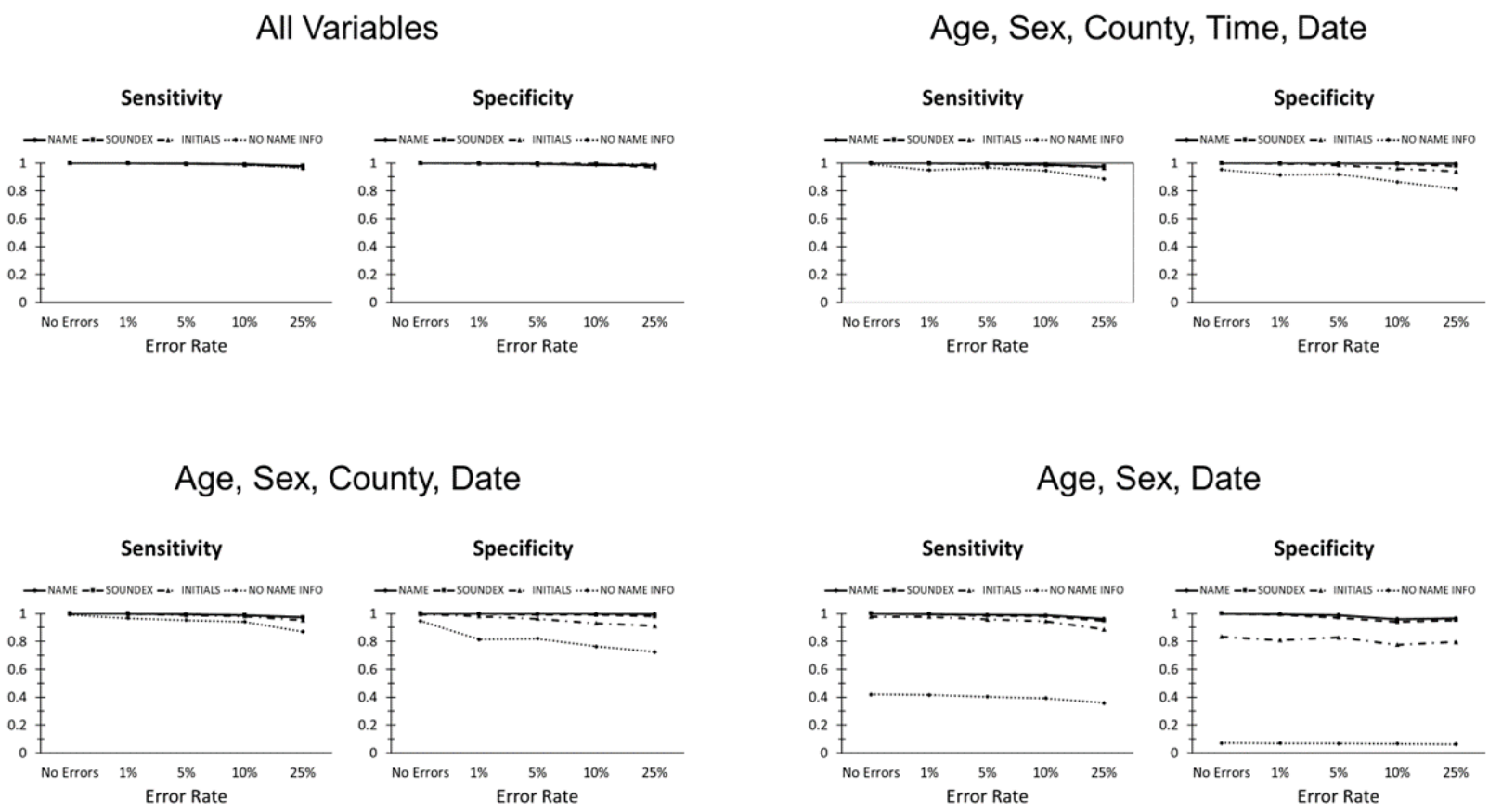

Figure 2. Sensitivity and specificity by type of name and non-name information available

The four graphs have a similar layout. The title of each graph provides the non-name fields that were used for this set of linkages. Sensitivity is shown on the left of each graph and specificity is shown on the right. Error rates are displayed across the x-axis and sensitivity/specificity on the y-axis. The results for linkages using full names are displayed using solid lines with a circle marker. Results for linkages using Soundex instead of names are presented with long dashed lines and a square marker. Results for linkages using first and last initials are presented with mixed dashed lines and a triangle marker. Finally, the results of linkages using no name information are presented with dotted lines and a diamond marker.

Starting in the top left, we can see that when linking with date of birth, sex, county, date, and time of crash/hospitalization, having full names adds little value to the results compared to linkages performed with no name information. In fact, the results from linkages using the four different levels of name information are nearly identical with sensitivity and specificity near one. The top right corner shows the results of replacing date of birth with age. For linkages using name, Soundex, and initials the results are fairly similar, again with sensitivity and specificity near one. The linkages using no name information are performing well, with sensitivity above 0.9 and specificity above 0.8, but the effect of introducing errors can be seen in this setting. As the error rate increases, both the sensitivity and specificity decline indicating it is becoming more difficult to distinguish between true and false matches. The bottom left corner shows the results from linkages with non-name fields of age, sex, county, and date of crash. In this setting, both linkages using full names and Soundex are still performing well. Linkages using initials are also performing well, but sensitivity declines with increasing error rate. The effect of having no name information is starting to become evident, with those linkages having a specificity below 0.8 for even modest error rates. Finally, in the bottom right corner are linkages performed using only age, sex, and date of crash as the non-name fields. Again, linkages using full names and Soundex have sensitivity and specificity close to one. The specificity for linkages with initials has declined to around 0.8 for all error rates. Finally, linkages with no name information struggle to distinguish between true and false matches with sensitivity below 0.5 and specificity below 0.1.

Can probabilistic linkage be successful without names? From these results we can see that with enough non-name information, names become redundant and do not impact the results. However, if you are in an information poor setting, then name information becomes critical.

Finding Hidden Information

The previous example has hopefully emphasized the importance of incorporating all possible information into a linkage model. While fields such as name, date of birth, sex, and date, time, and location of incident are obvious candidates for a linkage model, most databases have fields that do not seem immediately useful but include details that can increase information available for your linkage project. When working with ED or inpatient hospital discharge data International Classification of Diseases, Tenth Revision, Clinical Modification (ICD-10-CM) codes can provide useful information. Below are a few example codes and the types of information which may be useful to include in a linkage.

In our planned linkage of the MVC and ED databases a cause code of V20.4 allows us to determine a person was in a MVC and was the operator of a motorcycle. This information can be used to match with the vehicle type and seating position variables from the MVC database. In our linkage of ED to Poison Control, finding a diagnosis code of T39.012A identifies the record as being a self-harm/intentional poisoning visit that is related to aspirin ingestion. We can use this information to match against the substance, route, and intentionality fields in the poison control database. Finally, an injury code like S02.109A allows us to create body and nature of injury variables in both the ED and death certificate fields which may be useful when linking the two files.

Table 5. Hidden information in ICD-10-CM cause codes

V20.4: Motorcycle driver injured in a collision with pedestrian or animal in traffic accident	
Mechanism of injury	Motor vehicle crash
Crash Type	Motorcycle vs. Pedestrian/Animal
Vehicle Type	Motorcycle
Seating Position	Driver
T39.012A: Poisoning by aspirin, intentional self-harm, initial encounter	
Mechanism of Injury	Poisoning
Intent	Intentional/Self-harm
Substance	Aspirin
Date of event	This is the initial encounter
S02.109A: Fracture of the base of the skull, unspecified side, initial encounter	
Body Region	Head
Nature of Injury	Fracture
Date of Injury	This is the initial encounter

Knowing the location of the hospital where care was provided can be helpful even when the corresponding database does not include a hospital identifier, such as with the poison control database. For example, knowing a person was treated at the University of Utah Hospital tells us they were treated in Salt Lake County and Salt Lake City and the zip code was 84132 and the latitude and longitude was 40.77 N, 111.84 W. All of which may be of use to compare against the address of incident.

Data Cleaning

Another important step to take before building your linkage model is data cleaning. A careful review of all fields should be undertaken to ensure default and erroneous values have been set to missing, fields are coded the same in each database, and special data processing has been conducted. Many data systems provide default values to speed data entry. If persons entering the data use defaults when the true value is unknown then a pair of records may be unfairly penalized for a disagreement when the truth was not known. Examples of default values are ages of 0 or 99 and time of day of midnight. Other defaults may occur as a coding practice rather than being implemented in the collection software. For

instance, when the year of birth or age of a person is known but not their day of birth, an apparent ubiquitous practice is to assign a birthday of January 1st. A simple ordering by frequency of the birth date field should reveal if this is the case in your data. Similarly, if the hour of an event is known but not the minutes a common practice is to record times in the format of 1:11, 2:22, 3:33, etc. Again, a simple frequency of the most common times should reveal if this practice exists in your data. In situations where default values exist it is best practice to replace the default with a missing value. In cases of birth date, it may be possible to use birth year as a matching variable while making the month and day missing or to calculate the person's age rather than using their birthday. In our example with time of day, it may be useful to retain the hour of the event and discard the minute field if this practice appears to be widespread.

Ensuring fields are coded the same on both files before starting your linkage will prevent erroneous disagreements between pairs of records and save time reprocessing and rerunning your match. An example of when you need to recode occurs when one database has counties coded numerically 1 through 29, while the other database a string for the county name. In some cases, a field may have the same values in both databases but the labels for the values are different. For instance, one file may have counties coded 1 through 29 with labels in alphabetical order, while the other file has counties coded 1 through 29 with the labels organized by health district.

A final thought should be given to any special processing that needs to be done. Sometimes one database will have zero padding in numeric fields, e.g., "05," while the other does not. Sometimes special characters within names and strings are inconsistently entered. Last names such as O'Connell may be entered with or without the apostrophe. To prevent unintended disagreements in situations like this it is often helpful to pre-process name fields to remove apostrophes, hyphens, and other symbols. Remember, the goal of these modifications is to prevent erroneous disagreements between pairs of records and provide the best opportunity for true matches to be identified. We are not trying to correct or permanently change the data in the underlying databases.

Summary

Hopefully this section has impressed on you the importance and care needed when selecting variables for a linkage model. Before beginning your linkage, ensure you have removed default values from your fields, you've mapped data in both databases to the same coding scheme, and you have processed any fields that may have special characters. Also, conduct a careful review of all fields in the database to ensure you are not leaving any extra information out of your linkage, such as what can be derived from ICD-10-CM codes. Having this information available might make the difference in finding or missing true matches, such as when names are not available.

As we saw when discussing match weights, there are many properties of linkage variables that should be considered. The higher the reliability, the higher the probability of agreement for a true match. This results in larger *m* probabilities and higher match weights. The more levels a linkage variable has, the better the discriminating power and lower the probability of agreeing by chance on false matches. This

results in smaller *u* probabilities and higher match weights. Combining these two properties, the fields that frequently perform the best in probabilistic linkage models have more than a few levels, have observations distributed evenly across levels, have few missing observations, and are coded accurately.

Reviewing our example MVC crash database, the top tier candidate variables for our linkage are likely to be the date and time of crash, first and last name, and date of birth. Each of these fields we need to review for missing values and patterns of default coding. For data processing we might decide to create birth day and year of birth fields and change all 0101 birthdays to missing. We may also decide to remove apostrophes and hyphens from the last name field. Finally, reviewing the fields available on the example ED database, we see that having the full time of the crash may not be useful as the ED only captures hour of arrival, so we will remove minutes from the time field. Second tier variables might include county of crash, vehicle type, person type, and sex. Planning to use vehicle and person type means we will need to derive this information from the cause code in the ED database. If cause coding is poor, these fields will frequently be missing and vehicle and person type become less ideal matching fields. You may find it helpful to take a moment to write down initial variable rankings and data processing steps needed for the Poison Control to ED and ED to death certificates linkages.

Building a Linkage Model

After identifying and processing the fields for your linkage you are ready to start defining your linkage model. While this may seem like an obvious step of pairing like fields between your databases there are several key concepts to consider. When doing event-based linkages, such as a motor vehicle crashes or calls to poison control, it is important to have a mix of event and person information in your model. A model with too much event information will risk overmatching records for people involved in the same crash or vehicle. A model with too many person-level identifiers risks matching records for the same person across different events, for example a person with multiple crashes or ED visits in the same year.

All variables in a linkage model should be independent, meaning each one is providing unique information about the person and event. The most obvious dependent fields in our example databases are date of birth and age. Once the date of an event and date of birth are known then the age is completely determined. Similar dependencies can occur with hospital identifier, county of event, and zip code of event. If a hospital is known then the county and zip of the hospital are also determined. Other dependencies may not be so obvious. A person's last and first names are somewhat dependent. This can be due to geographic or ethnic differences. Several strategies can be employed to reduce dependencies in matching fields. For instance, while a person's full date of birth and age are dependent, their month and day of birth is independent of their age. Using only a portion of the first and last name, such as initials or Soundex of the first and last name usually eliminates the dependencies in these two fields.

After identifying the variables to include, the next step is to decide how you will compare them. Most linkage software will allow for a variety of comparison methods. Numeric variables may be compared as exact agreement or allowed to differ by a certain threshold. For example, if there is uncertainty in the accuracy of ages then one may choose to allow a one-, two-, or five-year difference between ages.

Similarly, a researcher may decide to allow the ED visit to occur up to three days after a MVC to account for persons who delay seeking care. Allowing for a difference in times is also common. There are several string comparisons methods that allow for small misspellings to be counted as agreements. While employing these tolerances is useful in helping account for noise in the data and provided a greater chance that true matches are identified, tolerances also provide false matches a higher probability of agreeing by chance. As an example, we will focus on day of the crash. Let us assume we would like to compare the date of an MVC and date of an ED visit requiring exact agreement. Because there may be inaccuracy in the way dates are collected and a percentage of persons in MVCs do not seek care on the same day of the crash, we set our *m* probability at 0.80. Thus, we expect the date of event to be the same date in 80% of all true matches. Assuming there are 365 days in a year and events are distributed evenly across days, then the probability of agreeing by chance on false pair of matches is 1/365 = 0.0027. If we are unhappy that up to 20% of true matches will receive a disagreement weight due to requiring an exact match on date, we could allow a one-day difference between the MVC and ED events. In this situation, let us assume that 95% of all true matches have the date of MVC and ED visit within a day. Now, because each date will have two days that are counted as matching, the probability of agreeing by chance is 2/365 = 0.0055. Similarly, if we wanted a larger window for agreement, say up to three days after the MVC, then we might find 98% of true match pairs now agree on date. It is now even easier for false match pairs to agree by chance, with the probability of agreeing by chance increasing to 4/365 = 0.0110. The resulting match weight, *w*, for each of these scenarios can be seen in the table below. Because increasing the date range increase the chance of false matches agreeing, the agree weight decreases as the range increases. Alternatively, because the *m* probability is increasing with increasing date range, pairs that do not fall within the range are penalized more heavily, with the disagreement weight more than doubling between zero and three days.

Table 6. Agree and disagree weights by adding a range to date comparisons

Difference in Days	m probability	u probability	Agree Weight	Disagree Weight
0 Days	0.8	1/365 = 0.0027	8.19	-2.32
1 Day	0.95	2/365 = 0.0055	7.44	-4.31
3 Days	0.98	4/365 = 0.0110	6.48	-5.61

Unique Identifiers

Sometimes the two databases contain a unique identifier. This can happen when both contain SSNs, or in cases where both databases use the same number as the unique identifier, like when joining MVC to citations and each file contains driver license numbers. Unique identifiers can cause instability in linkage models. Theoretically, this is because the probability that two people have the same SSN is zero, with results in a *u* probability of 0. This shows up in practice as extremely large match weights as we saw in our example above. Because of this instability, it is frequently better to directly join the two databases

using the unique identifier first. After you have removed all pairs identified this way, you can proceed to probabilistic linkage on the remaining records.

Example Linkage Models

We now develop example models for our three linkages. In our model for linking the MVC to ED databases (Table 7) we have decided to eliminate dependencies between the first and last name by using Soundex. We have also opted to divide the birthdate information into month and day of birth and age. By treating birthdate in this fashion, we can account for situations where a default birth day was entered. We are still able to incorporate information for records where the birthdate was missing but an age was recorded. We have also incorporated a one-day tolerance in date of the event to account for situations where the crash is near midnight or a person waits a day before deciding to seek treatment. Similarly, we have incorporated a four-hour window for comparing hour of the crash to hour of ED arrival to account for transport time. A difference of two years in age between the crash and ED records will also be allowed. Asterisks have been added to vehicle and person type to indicate they were derived from cause codes. Note that our model does a good job balancing event information (date and hour of MVC and hospital number) with person information (Soundex of first and last name, birthday, age, sex, and vehicle and person type).

Table 7. Model for linking the MVC and ED databases

MVC Database	ED Database	Comparison Method
Date of MVC	Date of ED Visit	+ 1 day
Hour of Crash	Hour of ED Visit	+ 4 hours
Soundex of First Name	Soundex of First Name	Exact
Soundex of Last Name	Soundex of Last Name	Exact
Month and Day of Birth	Month and Day of Birth	Exact
Age	Age	+/- 2 years
Sex	Sex	Exact
Vehicle Type	Vehicle Type*	Exact
Person Type	Person Type*	Exact
Hospital Number	Hospital Number	Exact

Let us now look at how we might approach a linkage between the ED and death certificate databases to identify deaths that occur following an injury-related ED visit. This first example assumes we are interested injury related deaths. As with the MVC and ED database we have chosen to use the Soundex of the first and last names to remove any dependencies between the two fields. Because the ED database is generated as part of billing and the death certificate is part of the public record, we are expecting this field to be coded very accurately. Therefore, we are treating the date of birth as a string and allowing one typo (a one-digit data entry error). Our belief in the accuracy of date of birth is represented by the *m* probability of 0.99, indicating we expect the birth date to match within one typo on 99 out of 100 true match pairs. We are matching the billing zip code with the decedent's zip code. Because a person may use a PO Box or have a billing address different from where they live, we are not

as confident in this field and have applied an *m* probability of 0.8. As we are doing an injury-related linkage we are choosing to match the date of the ED visit to the date of injury rather than the date of death. Because date of injury may be self-reported, we are choosing to allow a two-day tolerance on either side of the ED visit. We have also chosen to use an *m* probability of 0.85, indicating we do not expect it to be as reliable as names or dates of birth. Using date of injury is purely a decision to be made by the person performing the linkage and should be based on its completeness and accuracy. One method for checking the completeness of injury date is to calculate the percentage of records with an injury or cause code that have an injury date recorded. Talking with someone from the vital records office can be a good way assess the accuracy of the date of injury field. Finally, we have created a Cause of Injury field based on the ICD-10-CM codes in the ED database and ICD-10 codes contained in the death certificate database. The number of categories and their definitions are up to the person performing the linkage and likely depend on the size of the files and how common each cause is. One possible categorization could be: MVC, Fall, Burn, Poisoning, Other. We have given this field an *m* probability of 0.7, indicating we expect it to agree 70% of the time on true matches. Because of the lower *m* probability, a disagreement on the Cause Category will not ruin a pair's chance to become a true match if there are enough agreements on the other fields. There are many other possible ways of incorporating the information contained in ICD codes into this linkage. Rather than a categorization, one could choose to create indicator flags for each cause. Another possibility is to incorporate an intentionality categorization, such as unintentional, assault, self-harm, undetermined. Also, one may choose to experiment with body region indicators based on ICD-10-CM and ICD-10 codes. Whether or not these are useful and improve the quality of the linkage results will have to be determined on a case-by-case basis.

Table 8. Model for linking ED and death certificate databases related to an injury event

ED Database	Death Certificate Database	Comparison Method	*m* Probability
Soundex of First Name	Soundex of First Name	Exact	0.95
Soundex of Last Name	Soundex of Last Name	Exact	0.95
Date of Birth	Date of Birth	String: One typo	0.99
Sex	Sex	Exact	0.99
Billing Zip Code	Decedent's Zip Code	Exact	0.8
Date of ED Visit	Date of Injury	+/-2 days	0.85
Cause Category	Cause Category	Exact	0.7

Now let us consider a linkage between the ED and death certificate with the goal of identifying a death that occurs within a year of an injury-related ED visit. An example of such a study may be trying to identify the risk of death following an ED-treated suicide attempt. We are handling the person information in the same way as our injury-related linkage above. Because we are no longer focused solely on injuries, though, we are now comparing the date of the ED visit to the date of death. Also, because the we are interested in deaths that occur within one-year, we are allowing 365 days to pass between the ED visit and the death. Therefore, have also increased the *m* probability for the date

comparison to 0.99, indicating we expect almost all deaths to fall within this window. Also remember that by allowing a 365-day window we will be receiving a much lower agreement weight compared to our linkage above where we only allowed a window of plus or minus two days. For this linkage we have also created a flag to indicate if the visit and death are illness or injury related. Just as before, there are many different options for how to incorporate the information in the ICD-10-CM and ICD-10 codes and this is only one option.

Table 9. Model for linking ED to Death Certificates databases within one year of ED visit

ED Database	Death Certificate Database	Comparison Method	*m* Probability
Soundex of First Name	Soundex of First Name	Exact	0.95
Soundex of Last Name	Soundex of Last Name	Exact	0.95
Date of Birth	Date of Birth	String: One typo	0.99
Sex	Sex	Exact	0.99
Billing Zip Code	Decedent's Zip Code	Exact	0.8
Date of ED Visit	Date of Death	+ 365 Days	0.99
Injury/Illness	Injury/Illness	Exact	0.8

Blocking

Many linkage software products expect your linkage to be arranged in blocks. Blocking increases computational efficiency by reducing the pairs of records being compared to only those with a higher likelihood of being a true match compared to the universe of all potential pairs. To get a grasp on the concept of blocking, let us consider matching the MVC and ED databases. In our example we will assume there are 100,000 records in the MVC database and 1,000,000 in the ED database. To identify each MVC record's best match we have to compare it against all 1,000,000 ED records, resulting in 100,000 x 1,000,000 = 100 billion comparisons. Because most people do not seek ED care following their MVC, we may only expect 10,000 of the MVC records to match. That means we are making more than 99 billion comparisons we do not expect to result in a match. Blocking is a means to reduce the number of rejected comparisons. Rather than comparing the entire MVC database with the entire ED database, we instead will compare only pairs of records where the date of crash and ED visit are the same. If pairs of records are distributed evenly across all 365 days, we have reduced the number of needed comparisons to around 270 million (1/365th of one billion). We could reduce this further by only comparing pairs of records where the birthday and last initial are the same. If last initials are distributed equally, we have cut the number of potential comparisons to 10.5 million. It is tempting to continue in this fashion by also requiring the first initial to agree and reducing the expected pairs to only 400,000, however, the more fields we force to agree the more likely it is we will exclude true matches that happen to disagree on either the date of the event, the last initial, or the first initial. To address this potential pitfall, one should create multiple blocks. In our example, we may choose a second block where only records that agree on birthday and county of crash/ED visit agree. A third block might compare only pairs that agree date of crash/ED visit and hospital identifier.

Defining blocks should be done carefully. A good block contains both event- and person-level information. This will help reduce the chance of cross-matching all records from people involved in the same crash, in the case where only event information is included in the block or matching a MVC record to every ED visit a person has had, in cases where a block only contains person-level information. How restrictive blocks should be is dependent on how probabilistic linkage is implemented in your software. If your software removes pairs identified as matches from consideration in subsequent blocks then it is advantageous to begin with very restrictive blocks that will likely ensure the "best" matches are identified first. Subsequent blocks can then be created to cast a wider net to include pairs of records that didn't make it into the earlier blocks. For example, the first block could only compare records which agree on the Soundex of first and last name, age, and date of crash. Pairs identified here should be extremely likely to be true matches. The second block might only compare records that agree on first and last initial, and date of crash. The third block would then only compare records agreeing on date of birth, hour of crash/ED admission, and county of crash/hospital. Our fourth block would be the least restrictive of age and date of crash. In this fashion we create an inverted funnel where successive blocks gradually compare more and more pairs, but only after the pairs that agreed on the most stringent criteria were identified and removed from subsequent blocks. If, on the other hand, your linkage software allows all pairs to be compared in every block then it makes sense to identify blocks that are of approximately equal sizes. In this scenario we might consider the following four blocks: 1) last initial and date of crash; 2) date of birth and county of crash/ED visit; 3) age and date of crash; and 4) Soundex of first name and month of crash/ED visit. After all four blocks are complete your software will combine the results and produce a set of best matches.

In addition to ensuring individual blocks have a mix of event- and person-level information, one should take care to ensure all your blocks are not dependent. Dependent blocks can occur when the same variable is used in each block. If we were to include the date of crash/ED visit in all four blocks then we would never compare pairs where the clock crossed midnight between the two events or a person waited a few days prior to seeking care. Another way blocks can be dependent is if each contains information derived from a single field. This type of dependency can occur when every block relies on the date of birth or age, when age is calculated based on the dates of the event and birth, or when developing three passes with one pass using last name, one using the Soundex of the last name, and one using last initial. In these cases, a single error in date of birth or last name can result in a true pair never being considered.

Not all fields make ideal blocking variables. Sex has only two outcomes, which will only reduce the number of comparisons by half. Fields that are unreliable or have high levels of missing values are also not well-suited to be used as blocking variables. Also note that some of our blocks above use fields that we did not incorporate in our earlier linkage model. This is perfectly ok as our goal with blocking is to reduce the number of comparisons being made and eliminate likely false match pairs. Our linkage model is what will be used to identify true matches. So, blocking variables do not have to be included in your linkage model. Also, it is appropriate to use variables from your linkage model, like date of crash, age, and Soundex of last name, as blocking variables. Defining blocks and linkage models are separate processes.

Identifying Matched Pairs

Once you have run all your blocks and identified potential matched pairs you must decide which pairs are true matches and should be retained for analyses vs. which pairs are false matches and should be rejected before proceeding. The method for making this decision can be dependent on your software as well as your analytical needs. Most probabilistic linkage software will produce pairs with match weights. The higher the weight, the higher the likelihood that a pair is a true match. Depending on how weights are calculated you may be able to transform them into match probabilities. Having access to match probabilities allows you to utilize more sophisticated methods for identifying true matches. We will begin our discussion in the general setting with just match weights and then proceed to methods that utilize match probabilities.

Graphical Method

A very useful method for reviewing the results of your linkage is to make a histogram of your match weights. Under ideal conditions with good linkage variables, the histogram will be bimodal. On the left side of the histogram will be a large number of pairs with very low weights. When you examine these pairs individually, you should be able to immediately decide that they are not true matches and only happened to get identified because they agreed on one or two variables, like event date and county. On the right side of the histogram, you should see a smaller mode that consists of all the true matches. When you review these pairs, they should look like true match pairs with agreements on all or nearly all match variables. Between the modes there should be very little to no overlap. When you see this type of histogram you can pick a point between the modes and declare every pair above that mark a true match and every point below it a false match. The top row of the figure below depicts the ideal situation just described. The left side of the figure shows the histogram with pairs labeled by whether or not they are a true match. Because the databases do not share a common unique identifier, a researcher never knows which are true and false pairs. Instead, as we have seen, linkage provides a means of organizing pairs according to patterns of agreements and disagreements. While the left column of graphs is useful for understanding how the graph on the right arose, in practice we only see the right column, therefore these graphs will be our focus. With the labels removed we can still see clear separation between the mode of false pairs on the left and the mode of true pairs on the right. In this situation, we can pick any weight between 11 and 15 and declare all pairs above that value are true matches and reject the rest.

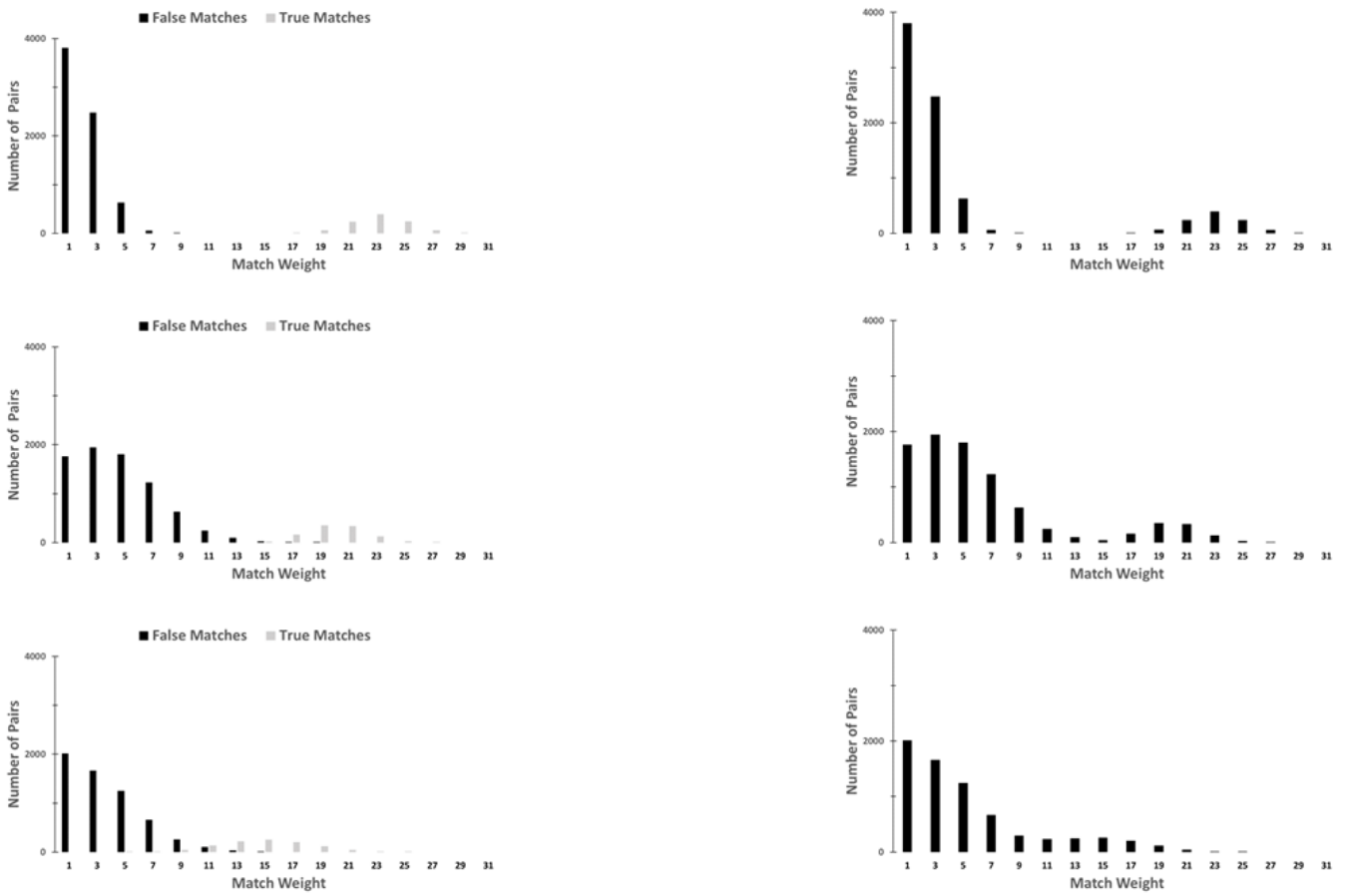

Figure 3. Example match weight histograms

The middle row shows a more common situation where there is some overlap between the true and false matches. Here the majority of the false pairs fall below a weight of 15 and a majority of the true pairs fall above a weight of 19. Unfortunately, we do not know the labels in practice and are faced with the histogram in the middle right. Looking at this histogram we see there is clearly some overlap between the two distributions between 11 and 17. There are several choices you can make here. It is always useful to inspect your pairs by match weight. We could begin our inspection at pairs with a weight of 19 and work our way down the distribution to pairs with a weight of 11. In the course of this review we may decide that there is a clear transition point from all or mostly all matches to all or mostly all false matches. If we identify this pattern then we can pick one point, as above, and declare all pairs above that weight true matches and all pairs below that weight false matches. Sometimes upon inspection it seems true and false matches are interleaved throughout the range from 19 to 11. In this case, it may be useful to manually assign pairs to be true and false matches as you are reviewing. While manual review can feel comforting by allowing you to weed out pairs you think are false matches, there is also a chance for you to unintentionally insert your own biases into the review process. For this reason, manual review should be undertaken with great care and caution. It may also be helpful to have a second person review the pairs to ensure agreement in the labeling of true and false matches.

The bottom row of the figure depicts a situation we never hope to see and is usually the result of trying to conduct a linkage without enough information to accurately identify the true matches. Looking at the graph on the left, we can see the distribution of true matches overlaps with a long tail from the distribution of false matches. While we have the bars shaded according to the truth it looks promising that we should be able to identify where the true matches begin and false matches end. Unfortunately, in practice, we are faced with the graph on the right, which looks unimodal with a long, right tail. This is typically a challenging situation because the linkage algorithm doesn't have enough information to adequately separate the true and false pairs. If you are faced with this histogram, it is important to verify you have included all the information available to you. Make sure that you have utilized ICD-10-CM codes and any other bits of data that might be contained in your database. You may also want to check with the data owners to see if there are additional fields that may be of assistance.

High Probability Pairs

If match weights are calculated as described in our earlier section using *m* and *u* probabilities, then it is possible to transform the match weight into an overall probability that the pair is a true match. This method has been described by several authors and is beyond the scope of this guide.[21-24] To make these calculations, you will also need to supply your software with the number of pairs you expect your linkage to return. Using results of a previous linkage is one way to estimate the number of expected pairs. If this is the first time you are conducting this linkage then you can usually obtain estimates from fields in your databases. For instance, the MVC database has a field that identifies if a person was transported by EMS, which may be useful for counting the expected number of matches to the ED database. Alternatively, cause codes in the ED database can be used to estimate the number of persons treated for injuries sustained in motor vehicle crashes. Likewise, the poison control database contains a field that identifies whether a person was referred to the ED.

Once you have the match probability you can proceed in a number of ways. One method is to pick a cutoff probability, such as 0.9, and declare any pair with a probability above 0.9 is a true match and all others are false matches. This is often referred to as creating a set of high probability matches. When creating a set of high probability matches, it is often best practice to create the histograms discussed above but using match probability for the x-axis instead of match weight. This will allow you to see where the majority of your matches fall. For our examples above, we will assume that a match weight of 19 is equivalent to a match probability of 0.9. In the first row, selecting a probability of 0.9 would keep almost all true matches but miss a handful of pairs at 17 and 18, while rejecting all false pairs. In the middle row a match probability of 0.9, weight = 19, would again identify the majority of true matches. However, we would end up rejecting more than just a few true pairs with probabilities below 0.9 (match weight < 19). In this situation we would let a small number of false matches become matched pairs. If you are faced with a histogram like the middle row, it may be useful to create one cutoff point at a probability of 0.9 and another at a probability of 0.5. Pairs with a probability above 0.9 will be declared true matches and then hand review will be conducted on pairs with a probability between 0.5 and 0.9. The same cautions about hand review introducing bias should be remembered here.

Selecting a match probability of 0.9 (match weight = 19) for the graphs in the bottom row is not useful. We see this would only result in a minimal number of pairs being identified as true matches. Further, it is likely that pairs that were able to make our strict criteria of a match probability of 0.9 are highly biased. They likely have the rarest values of most of our matching variables. For instance, these pairs may represent older driver crashes in rural counties occurring between midnight and 5:00 am. More common events, such as teenage crashes in urban areas would not contain enough information to achieve a match probability of 0.9. Here, as stated above, the best solution is to inspect your databases for fields you may have forgotten to include in your matching algorithm or ask the data owners if there are additional fields to augment your current data set.

Controlling the False Match Rate

Another option for creating a set of match pairs when your software provides the match probability is to control the false match rate. The first step in this process is to order your pairs in descending order of match probability so the highest probability is first and the lowest probability is last. For each pair calculate the probability that it is a false match by subtracting the true match probability from 1. For instance, if your highest match probability is 0.999 then the probability the pair is a false match is 0.001. Then beginning with the highest probability pair add it to the set of true matches and calculate the false match rate as the sum of the false match probabilities divided by the total pairs. In our example, we would add the pair with a probability of 0.999 and then calculate the false match rate as 0.001/1. This would give us a false match rate of 0.1%. Next, we add the second highest probability pairs and then calculate the new false match rate. The algorithm continues in this fashion until the *a priori* false match rate is achieved. Commonly selected false match rates are 1% or 5%. This method tends to work well when you have situations there is enough information for the linkage algorithm to produce a histogram with separation between the true and false pairs, as depicted in the first two rows of our match weight histograms.

Imputed Matched Sets

When your databases do not contain enough information to adequately separate the true and false matched pairs the methods discussed above produce matched sets that are likely biased and exclude the majority true matches. If there is not additional information to add to the linkage algorithm then imputed matched sets may be a useful solution. Imputed matched sets generate multiple sets of matched pairs weighted on the true match probabilities. Pairs that have a true match probability of 0.9 will be selected in about 9 of 10 samples. Pairs with a true match probability of 0.1 will only be selected in about 1 of 10 samples. Rather than creating a single set of matches, we repeatedly sample and create multiple sets. While this method will result in some pairs with very low match probabilities being retained as true matches and some pairs with high match probabilities being rejected as false matches, it has been shown imputed match sets produce distributions of the matching fields that are more similar to the truth than relying on high probability pairs alone. [25, 26] Utilizing imputed matched sets increases the complexity of your analyses and may result in the need for utilizing specialized software to combine the results from multiple data sets. [27, 28]

Evaluating Linkage Results

Whichever method you choose for generating your final match pairs you should evaluate your results. The first item that is useful to check is: Did the number of matched pairs come close the number you expected? If you found many fewer pairs than expected it may be useful to manually review your database to see if you can identify missed pairs. For instance, you may wish to create a data set of unmatched records from the MVC file that have high injury severity codes and are listed as having been transported to the ED. From here you can sort your ED file by date of admission, date of birth, or age and see if you can find these missed matches. If you can locate several matches through this method, you can often identify an error in your matching algorithm or blocking scheme that prevented the pairs from being compared or being given a high match weight or probability. Sometimes this type of review will not yield any new matches, which means you should update your expected number of matches for the next time you perform this linkage. Rather than expecting all ED records with a cause code of an MVC to match, you might instead expect 80% to match.

Once you have results this is a great time to evaluate the reliability estimates, *m* probabilities, you supplied your software at the beginning of the linkage process. Among your set of matches calculate the percent agreement on each of the matching fields. This can be done by creating an indicator variable for each field. For example, to create an indicator for age, you will assign a value of 1 if the MVC and ED ages are the same (or fall within your specified tolerance), a value of 0 if the ages disagree (or are outside your specified tolerance), and a value of missing if age is missing on one of the two files. Then simply calculate the percentage of pairs that have a value of 1. If you identify match fields that have agreements much higher or much lower than your initial estimate, you will want to make a note to update these values when you perform this linkage next time.

A final check is to review your pairs from highest match weight to lowest match weight to ensure the patterns of agreement and disagreement lead you to feel confident these are true matches. If you see unexpected results, it can often be the result of a miscoding of one or more linkage fields.

Reporting Linkage Results

When reporting your linkage results you should provide details about the linkage process so others can assess your methods and results. The amount of information to report likely depends on the setting where you are presenting. The most important information to provide your audience is which fields you included in your matching process. You should also describe how you selected your final matched pairs. Did you use a histogram or match probabilities to select a high probability set? Did you do hand review or stick with a single cutoff value? Did you use imputed matched sets? And why did you select the method you used. How many matches did you identify and was this close to the number you expected? For more technical audiences it is often useful to present the mean, median, and the 25^{th} and 75^{th} percentiles of your probabilities. Reviewers and others may also ask how many blocks you used and what variables were used in each block.

Probabilistic vs. Deterministic Linkage

The most straightforward method to combine databases is when you have a common key, such as SSN, that allows for standard database joins and operations to be performed. If you have data with common keys, you should use them. When a common key does not exist then you need to use deterministic or probabilistic methods to combine your databases. By now, you have seen that probabilistic linkage has many data cleaning and pre-processing steps, requires investigation and understanding of the properties of your data, and has a strong theoretical underpinning. An alternative approach to probabilistic linkage is deterministic linkage. In deterministic linkage, a researcher defines sets of variables they feel represent patterns that should exist on true matches. For instance, a researcher may decide all pairs that agree on first name, last name, and date of event are matches. Another form of deterministic linkage is to assign point totals to different fields, where fields that are more likely to identify a true match receive higher points than fields that are more general. For instance, a research may assign five points for agreement on each of first and last name, date of birth, and date of incident. County and hour of event will be assigned two points for agreement. Agreement on sex will only result in one point. The overall score for a pair of records will be the sum of all the agreement scores. The researcher will then decide a cut point, much like the graphical method above, for deciding which pairs are true matches.

Deterministic linkage may seem like an attractive option as the process is not as involved as probabilistic linkage. It is sometimes stated that if you have highly specific identifiers, like names, then deterministic linkage may be the preferred method. Before choosing deterministic linkage there are a few limitations to keep in mind. With deterministic linkage, you lose the ability to state your confidence in the reliability of a field as we do with *m* probabilities in probabilistic linkage. That means the penalty for disagreement cannot be moderated in deterministic linkage for less reliable fields. We saw above that lowering a field's *m* probability decreases the disagreement weight, therefore creating less of a negative impact on a pair's overall match probability. Thus, when fields do not have near perfect reliability, deterministic

linkage may fail to identify true matches. Similarly, in deterministic linkage an agreement is assigned the same weight regardless of the value. In probabilistic linkage we can assign value specific weights that are directly related to how rare the value is, with rare values receiving higher weights than common values. Value specific weights provide probabilistic linkage a greater opportunity to identify true matches compared to deterministic linkage. As we have seen, probabilistic linkage provides the probability that a pair of records is a true match. Having the match probability provides you the ability to make informed decisions on which pairs to retain for true matches. You are also able to provide justification for your decision to other researchers and journal reviewers. Whereas, with deterministic linkage you are have a set of matches that you have decided are right but have no metrics for describing the quality of the results. In our experience, probabilistic linkage always identifies the same matches as deterministic linkage, while also identifying pairs with some disagreements that are included due to value specific match weights and the ability to account for the reliability of each field. Therefore, while deterministic linkage may seem more straightforward, the advantages gained through probabilistic linkage makes it worth the effort and generally produces better results.

Summary

Clearly probabilistic linkage is not a method to be undertaken lightly. Probabilistic linkage is a procedure that relies on the statistical properties of fields in your database. While your software will most likely calculate agreement by chance, the discriminating power or *u* probability, it is incumbent on the researcher to supply the value for the reliability, *m* probability, for each field. When considering *m* probabilities remember to account for how the field will be compared and if a tolerance will be used. Each field will be assigned a match weight, *w*, based on the ratio of *m* and *u*. Rare values within a field are assigned higher weights than common values. Fields with high reliability receive a bigger penalty for disagreeing compared to fields with lower reliability. When defining your linkage model remember to avoid using dependent fields. Dependencies can be handled in a variety of ways. The first is to select a single field, such as when deciding between event zip, city, and county. Another option is to use transform fields to reduce or eliminate the dependency. An example transformation is using the Soundex of first and last name rather than the actual names. Another example is using the month and day of birth along with age rather than using the full birth date with age. Once you've set your probability model, you will likely need to define blocks. Remember, blocking simply provides a means to reduce the computational requirements by restricting comparisons to pairs of records that are more likely to be true matches than random sets of pairs. It is important to ensure that your blocks are not dependent, so you don't unintentionally exclude potential true matches. Once you have executed your matching algorithm you need to determine which pairs are the true matches to be carried forward into your analysis and which ones should be rejected. This decision can be based on a combination of graphs and reviewing match probabilities. Finally, make sure to provide reviewers and other researchers enough information to evaluate the results of your linkage. Information you should provide includes the fields used in your linkage, how you selected your final matches, and descriptive statistics for your match probabilities.

Data linkage is a powerful tool for injury epidemiologists to combine information from multiple databases. This how-to guide provides the tools and knowledge to successfully undertake linkage projects. With linked data, epidemiologists are better equipped to monitor trends, identify at-risk populations, and evaluate programs and interventions to help reduce the incidence and societal costs due to injuries.

References

1. Singleton, M.D., *Differential protective effects of motorcycle helmets against head injury.* Traffic Inj Prev, 2017. **18**(4): p. 387-392.
2. Curry, A.E., et al., *Motor Vehicle Crash Risk Among Adolescents and Young Adults With Attention-Deficit/Hyperactivity Disorder.* JAMA Pediatr, 2017. **171**(8): p. 756-763.
3. Olsen, C.S., et al., *Motorcycle helmet effectiveness in reducing head, face and brain injuries by state and helmet law.* Inj Epidemiol, 2016. **3**(1): p. 8.
4. Han, G.M., A. Newmyer, and M. Qu, *Seat belt use to save face: impact on drivers' body region and nature of injury in motor vehicle crashes.* Traffic Inj Prev, 2015. **16**(6): p. 605-10.
5. Makara, J., et al., *A cross-sectional study of characteristics of bicyclist upper and lower extremity injuries in bicycle-vehicle crashes in Ohio, United States, 2013–2017.* BMC Public Health, 2021. **21**(1): p. 428.
6. Zhu, M., S.B. Hardman, and L.J. Cook, *Backseat safety belt use and crash outcome.* J Safety Res, 2005. **36**(5): p. 505-7.
7. Burch, C., L. Cook, and P. Dischinger, *A comparison of KABCO and AIS injury severity metrics using CODES linked data.* Traffic Inj Prev, 2014. **15**(6): p. 627-30.
8. Cook, L.J., et al., *Crash Outcome Data Evaluation System (CODES): An Examination of Methodlogies and Multi-State Traffic Safety Applications*. 2015, U.S. Department of Transportation, National Highway Traffic Safety Administration: Washington DC.
9. Centers for Disease Control and Prevention (CDC) National Center for Injury Prevention and Control (NCIPC), *LInking Information for Nonfatal Crsh Surveillance (LINCS): A guide for integrating motor vehicle crash data to help keep Americans safe on the road*. 2018: https://www.cdc.gov/transportationsafety/linkage/index.html.
10. Wilton, J., et al., *Prescription opioid treatment for non-cancer pain and initiation of injection drug use: large retrospective cohort study.* BMJ, 2021. **375**: p. e066965.
11. Carlson, K.F., et al., *Linkage of VA and State Prescription Drug Monitoring Program Data to Examine Concurrent Opioid and Sedative-Hypnotic Prescriptions among Veterans.* Health Serv Res, 2018. **53 Suppl 3**: p. 5285-5308.
12. Ashraf, A.J., et al., *Receipt of Concurrent VA and Non-VA Opioid and Sedative-Hypnotic Prescriptions Among Post-9/11 Veterans With Traumatic Brain Injury.* J Head Trauma Rehabil, 2021. **36**(5): p. 364-373.
13. Kyerematen-Amoah, E., et al., *Public Health Surveillance for the Prevention of Pesticide-Related Illness in Illinois.* J Occup Environ Med, 2020. **62**(5): p. 359-369.
14. Cerel, J., et al., *Emergency Department Visits Prior to Suicide and Homicide: Linking Statewide Surveillance Systems.* Crisis, 2016. **37**(1): p. 5-12.
15. Chitty, K.M., et al., *Australian Suicide Prevention using Health-Linked Data (ASHLi): Protocol for a population-based case series study.* BMJ Open, 2020. **10**(5): p. e038181.
16. Wu, J., et al., *Record linkage is feasible with non-identifiable trauma and rehabilitation datasets.* Aust N Z J Public Health, 2016. **40**(3): p. 245-9.

17. Kumar, R.G., et al., *Probabilistic Matching of Deidentified Data From a Trauma Registry and a Traumatic Brain Injury Model System Center: A Follow-up Validation Study.* Am J Phys Med Rehabil, 2018. **97**(4): p. 236-241.
18. Durojaiye, A.B., et al., *Linking Electronic Health Record and Trauma Registry Data: Assessing the Value of Probabilistic Linkage.* Methods Inf Med, 2018. **57**(5-06): p. 261-269.
19. Chen, Y., et al., *Linking Individual Data From the Spinal Cord Injury Model Systems Center and Local Trauma Registry: Development and Validation of Probabilistic Matching Algorithm.* Top Spinal Cord Inj Rehabil, 2020. **26**(4): p. 221-231.
20. National Archives. *The Soundex Indexing System*. [cited 2021 12/24/2021]; Available from: https://www.archives.gov/research/census/soundex.
21. Cook, L.J., L.M. Olson, and J.M. Dean, *Probabilistic record linkage: relationships between file sizes, identifiers and match weights.* Methods Inf Med, 2001. **40**(3): p. 196-203.
22. Jaro, M.A., *Probabilistic linkage of large public health data files.* Stat Med, 1995. **14**(5-7): p. 491-8.
23. Fellegi, I. and A. Sunger, *A Theory for Record Linkage.* JASA, 1969. **64**(328): p. 1183 - 1210.
24. Newcombe, H.B., et al., *Automatic linkage of vital records.* Science, 1959. **130**(3381): p. 954-9.
25. Thomas, A.M., et al., *The utility of imputed matched sets. Analyzing probabilistically linked databases in a low information setting.* Methods Inf Med, 2014. **53**(3): p. 186-94.
26. McGlincy, M.H. *A Bayesian record linkage methodology for multiple imputation of missing links*. in *JSM*. 2004. Toronto, CA.
27. Little, R.J.A. and D.B. Rubin, *Statistical Analysis with Missing Data*. Wiley Series in Probability and Statistics. 2002, New York: Wiley.
28. Rubin, D.B., *Multiple Imputation for nonresponse in surveys*. 1987, New York: John Wiley & Sons.

CHECKLIST OF BEST PRACTICES FOR DATA LINKAGE IN INJURY EPIDEMIOLOGY

Getting Started

Prior to beginning data linkage

Partnership with stakeholders

- Build partnerships with the database owners to appreciate the needs of the stakeholders and the privacy policies for the database.
- A trustful partnership with the database owners and the entity of state-agency could allow the access to maximal data elements for data linkage.

Data use agreements and acquiring data

- Data use agreements are usually needed to access the datasets with personal identifiers.
- Be patient to go through the steps for data use agreements to acquire the datasets.
- Follow data use agreements to store the datasets securely. Make the datasets with personal identifiers only accessible to the authorized epidemiologists.
- Renew data use agreements on a regular basis.
- Utilize the source datasets and linked datasets for allowable use under data use agreements.

Data Linkage Process

Conducting the linkage project and analyzing data

Identify potential variables

- **Personal identifiers**
    - Social security number
    - Driver's license number
    - Name
    - Date of birth
    - Sex
    - Residential address
    - Further derivation from external causes of morbidity codes (E-codes) (e.g., person role and vehicle type)
- **Event identifiers**
    - Date of event (e.g., date of hospital admission)
    - Time of event (e.g., hour of hospital admission if possible)
    - Location of event
        - Exact address
        - Latitude / longitude
        - Broader city or county

Deciding on deterministic matching

- A unique single identifier (e.g., social security number, driver's license number)
- A combination of quasi-unique identifiers:
    - Combination of date of birth, sex, and name
    - Combination of name and residential address
- Data availability and quality often preclude deterministic matching.

Proceeding with probabilistic linkage

- Unique or quasi-unique identifiers are not available.
- Data quality issues (e.g., missing values and errors) with the unique or quasi-unique identifiers

Data preparations and variable standardization

- Often the most time intensive part of data linkage
- Examine the frequency, distribution, missing values, and errors of data elements. Pay attention to the values that are more frequent than the expected as this usually signals missing, unknown, or default values.
- Randomly select 100 records to print all the linkage data elements to appreciate the overall quality of the data
- Set missing values for numeric and character variables
- Derive variables (e.g., role type in motor vehicle crashes) from the external causes of morbidity codes (i.e., E-codes) if applicable
- Standardize the variable values between the data elements from two databases (e.g., 'F' for sex from police crash report database and hospital database)
- Generate a unique ID to identify observations from each database before data linkage. This unique ID is a computer-assigned case number and can be extremely helpful to retrieve data elements during and after linkage. It also protects data privacy.
- Create a linkage dataset to select only the unique ID and linkage variables to reduce the size of the data and improve the proficiency of the linkage.

Self-match to examine and remove deduplicate records

- This is particularly important for hospitalization database as the same individual can visit the hospital multiple times for the same medical reason.
- Linkage variables can be expanded to include more data elements. For example, principal diagnosis and primary payor may be useful to self-match hospitalization records.
- The cut-off probability to identify self-matches is typically 0.9.

Dual match: matching variables

- Include a mix of both event and person level variables for event-based linkages. Event-level variables include the data, time, and location of the event. Person-level variables include names, age, sex, residential address, and so forth.
- The E-codes can be helpful to derive variables for linkages in injury.

- Matching variables are evaluated for each candidate pair. If the matching variables agree on the value from the two databases, the agreement weight is added for the candidate pair. If the matching variables do not agree on the value, the disagreement weight is subtracted for the candidate pair.
- The probability for a match is estimated from the sum of agreement and disagreement weights for all the matching variables.
- While the exact same value is considered as agreement in many variables (e.g., sex), tolerances can also be allowed. For example, the tolerance can be set for one day between the date of motor vehicle crash and the date of hospital visit. Another example is Soundex match for name. Soundex codes names by sound so that minor differences in spelling can be allowed when matching names.
- If two matching variables have strong dependency or correlation, the variables need to be transformed or one variable needs to be removed. An example is using the month and day of birth along with age rather than using the full date of birth with age. Another example is selecting only one matching variable from event zip code, city, and county. If two dependent variables are kept in the dual match, a penalty for the agreement weight needs to be implemented for the two dependent variables.
- Matching variables which act as unique personal identifiers are associated with large uncertainty in matching weight. If possible, perform an inner join on the two databases using the unique personal identifiers. Then proceed to probabilistic linkage on the remaining records.
- Eliminate duplicates identified in self-match before dual matching.

Dual match: blocking variables

- Blocking variables are used to identify candidate match pairs.
- Blocking variables can be a matching variable but can also be a non-matching variable.
- Use a combination of event- and person-level variables (e.g., event date and individual's age and sex).
- Use at least two or three sets of blocking variables. Avoid using the same variable in each block.
- Blocking variables should have minimal errors.

Selecting matched pairs

- Make a histogram of the match weights. The histogram should be bimodal when the linkage model is good. When two modes of match weights overlap, manual review of records can be used to decide whether a match is true or not.
- When the histogram of match weights is unimodal with a long tail, the linkage model does not separate the true matches from false matches. Efforts need to be pursued to use all the available matching variables and information. If these efforts cannot help produce a bimodal histogram of match weights, the use of multiply imputed matched sets (Markov Chain Monte Carlo) is recommended. This process is analogous to multiple imputation for missing values.
- A single set of matched pairs can be selected based on user defined threshold, estimated number of matched pairs, or maximum of the posterior distribution of match probability.

- If the cut-off probability of 0.9 is used to identify matched pairs, some true matched pairs may be missed, and the identified matched pairs may not be representative of all true matches.

- The use of a single set of matched pairs may underestimate the true variability in the data and result in false significance from hypothesis tests and artificially small confidence interval widths.

Examining quality of dual match

- A random selection of 100 matched pairs can be used to examine the quality of dual match. The high match probability pairs (≥ 0.9) are expected to agree on most matching variables. The low match probability pairs are expected to disagree on most matching variables.
- Check if the number of matched pairs is close to the expected number. If the number of matched pairs is much less than the expected, a manual review of unmatched records may be helpful to identify missed pairs. If no new matched pairs can be identified, the expected number of matches might need to be set lower.
- Evaluate the reliability estimate (often denoted as *m)*, the percent agreement on each of the matching variables among the matched pairs. The reliability estimates might need to be adjusted for the next linkage if they are much different from the initial estimates.
- The distribution of matching variables in the source database and matched pairs may help identify whether certain subgroups are missed in the matched pairs. For example, a comparison can be made on the distribution of the role types (e.g., passengers) in the matched pairs vs the source police crash reports and hospital file. Please note that the difference in the distributions across the source database and the matched pairs can be influenced by both the availability of matching variables and the likelihood of the outcome. Suppose that a higher percentage of passengers are in the police crash report, relative to the matched pairs between police crash report and hospitalization records. This could be due to limited identifying information on passengers from the police crash report, or lower likelihood of hospitalization for passengers involved in crashes.

Analyzing linked data

- It is good practice to left join the source database with the matched pairs for the analysis. For example, keep all the records from the source database (e.g., police crash reports) and add the medical outcomes for those records that are linked with hospital records. This enables calculating the proportion of hospitalizations for all individuals involved in traffic crashes. Using only the crash records linked to a hospitalization does not allow this calculation.
- Creating an analytic dataset that removes all personal identifiers and keeps the unique ID (i.e., computer-assigned case number) and the variables that are potentially used in analysis. This would reduce the number of variables. Otherwise, the multiple medical diagnoses and procedures coupled with multiple visits for the same individual would result in many variables. You can always use the unique ID to retrieve more variables from the source database if needed.

- Follow data use agreements to store the datasets securely and conduct allowable analyses. Make the datasets with personal identifiers only accessible to the authorized epidemiologists. Broader access can be provided to the relevant study team for the analytic dataset without any personal identifiers.
- For analytic ideas, use variables from both source databases because this is the benefits and strengths of the data linkage. For example, examine the association between seat belt use among persons in crashes (identified from the source police crash reports) and medical outcomes in term of hospitalization charges, traumatic brain injuries, and length of stay (available from the source hospital records).
- Be aware of the data limitations. For example, if only high probability links are analyzed, the results may not be representative of the study population.
- Multiple imputation analysis techniques are needed if the multiply imputed links are used. However, this analysis is the same as the increasingly routine multiple imputation methods for missing values.

Dissemination

Communicating results and project documentation

Sharing linkage progress and findings with stakeholders and injury prevention community

- Organize regular meetings of stakeholders including data owners to share linkage progress and findings to foster a long-term partnership. The meeting frequency should be at least annually.
- Constantly provide feedback on data quality to data owners to improve data quality.
- Analyze the linked data to address the needs from the stakeholders.
- Disseminate findings broadly through one-page fact sheets, reports, and peer-reviewed journal articles. Present findings at stakeholder meetings and scientific conferences including but not limited to CSTE and Safe States annual conferences.

Documentation

- Documentation cannot be over emphasized. Document your linkage practice and programs. This will be quite helpful because the linkage is often done on a yearly basis, and you will forget many things after a year.
- Documentation helps train new linkage epidemiologists and facilitates other epidemiologists to replicate and verify linkages.

SURVEY OF SELECTED LINKAGE SOFTWARE

To help appreciate various software for data linkage by state, tribal, local, and territorial injury epidemiologists, we searched online for linkage software. Selected software was described according to the online documents and user's manuals as of December 21, 2021. Except for LinkSolv, other software was not tested. Our selected software was not a complete list for linkage. Additional software was identified in the National Center for Injury Prevention and Control's report entitled "Linking Information for Nonfatal Crash Surveillance (LINCS): A guide for integrating motor vehicle crash data to help keep Americans safe on the road" and the National Highway Traffic Safety Administration's report entitled "Crash Outcome Data Evaluation System (CODES): an examination of methodologies and multi-state traffic safety applications".[1,2]

LinkSolv

LinkSolv (http://www.strategicmatching.com/) is the commercial version of the linkage software utilized by the CODES Data Network previously supported by the National Highway Traffic Safety Administration.[2] The latest version is 9.1.1336 released in February 2021. The cost is approximately $5,000 including perpetual single-user license and one-year technical training and support to complete data linkages.

LinkSolv runs on Microsoft Access platform and can also connect to SQLServer to link big databases. LinkSolv uses the probabilistic methodology to calculate match probabilities. It can properly compare various type of matching variables including numbers, character strings, dates, times, latitudes/longitudes, and so on. Each comparison can be an exact agreement between the two matching variables or an agreement within a specified tolerance. For example, two numbers can be regarded an agreement if they are the same, or within a numeric distance or percentage. Many built-in standardization routines help users prepare data files for linkage within the software, rather than doing so before importing the file. For example, Soundex can be used to standardize names by sound, so that minor differences in spelling can be allowed when matching names. Also, LinkSolv has a standardization routine to derive information from E-Codes (external causes of morbidity) of ICD-9 and ICD-10 diagnostic codes. Injury epidemiologists can also customize standardization and comparison rules.

LinkSolv can simulate data to help users to build linkage models. With simulated data, users know which matched pairs are true matches a priori. The linkage of the simulated data helps epidemiologists estimate the fit of the linkage model and the percent of ascertained true matches. Using the simulated data, an injury epidemiologist can develop well-performing linkage specifications. Then he/she can apply the linkage specifications to the real data.

LinkSolv up-weights and down-weights a specific value of matching variables given its frequency. It also assigns a match probability to each matched pair.

LinkSolv offers several ways to select matched pairs. One way is to use 0.9 as the cut-off probability to select high-probability matches. Another way to produce a single set of matched pairs is to rely on the

maximum of the posterior distribution of match probabilities from at least 50 imputed sets. The third way is to use Markov Chain Monte Carlo (MCMC) method to impute multiple sets of matched pairs. Although the MCMC and multiple imputation methods address the potential limitations of focusing on a single set of matched pairs, such as the underestimated standard errors from common probabilistic linkages, injury epidemiologists need to be familiar with the multiple imputation method to analyze the imputed sets.

LinkSolv provides tools to evaluate the quality of the linkage. One can examine the impact of widening or shortening the tolerances between matching variables. One can also evaluate if two matching variables (e.g., date of birth and age) have dependent agreements or disagreements. If a dependency or correlation is identified, one variable needs to be removed or a penalty for the agreement weight needs to be implemented for the two dependent variables. Furthermore, all match specifications and testing results are compiled in the reports to facilitate users to review linkage specifications and make modifications for future linkages.

LinkSolv has a function for a self-match to identify duplicate records (i.e., multiple records from the same person). Three databases can also be matched simultaneously (i.e., triple-match).

Match*Pro

Match*Pro (https://surveillance.cancer.gov/matchpro/) is a free probabilistic linkage software developed by Information Management Services, Inc. The latest version is 2.0.6 released in December 2021.

Match*Pro can import data files in fixed-width, delimited, or NAACCR XML format. The software provides the ability to validate matching various variables such as date, name, and address. Match*Pro allows for exact matching or matching within a specified tolerance, on date, name, zip code, SSN, address, and phone number.

Match*Pro offers several ways to define blocking variables to identify candidate match pairs including Soundex method for name and various combinations of year, month, and day for date. The software also provides several settings to calculate the agreement and disagreement weights. The additive setting counts agreement weights only, but does not deduct disagreement weights. The linear setting uses partial agreement weights for matching within a tolerance. The binary setting gives the full agreement weights when the matching exceeds the threshold. Both linear and binary settings use the full disagreement weight when the threshold is not passed.

Probabilistic weights can be calculated using the expectation-maximization algorithm or be user defined. The cut-off weight for matches can be user defined, or auto-adjusted based on the number of expected matched pairs. The software facilitates manual review of candidate matched pairs. Match*Pro does appear to help deduplicate files.

Match*Pro does not appear to assign a match probability to each matched pair. It cannot be used to simulate data for linkage purposes. The MCMC method is not available to refine the linkage model. Imputed matched sets cannot be created.

R: RecordLinkage

R: RecordLinkage (https://cran.r-project.org/web/packages/RecordLinkage/index.html) is a free R package. The latest version is 0.4-12.1 released in August 2020. As RecordLinkage runs on R platform, users need to utilize R to import and read data. As there are no built-in validation and standardization routines, R programming is needed to clean and standardize variables.

Matching variables can be matched exactly. Or matching variables can be set within a specified tolerance such as Jaro-Winkler string distance for names. The matched pairs can be selected based on a threshold determined by the user or a training dataset. Another option is the software-identified threshold based on extreme value statistics.

RecordLinkage has a feature to use machine learning methods such k-means clustering or bagged clustering instead of a probabilistic linkage model to deduplicate records and link databases.

RecordLinkage does not appear to assign a match probability to each matched pair. It does provide the ability to deduplicate multiple records from the same person.

RecordLinkage cannot be used to simulate data. It does not appear to have built-in algorithms to assess the fit of a linkage model. There are no tools to impute multiple matched sets.

R: FastLink

As a free R package, R: FastLink (https://cran.r-project.org/web/packages/fastLink/index.html) enables users to conduct probabilistic linkages that can incorporate auxiliary information. The latest version is 0.6.0 released in April 2020. Users need to be familiar with R to import and read data before linkage.

There are some standardization routines for name and address. FastLink is flexible in defining blocking variables to identify candidate match pairs. Blocking variables can be an exact match, match within a tolerance, or k-means blocking for names.

For matching variables used to calculate agreement or disagreement weights, exact match or match within a tolerance (e.g., Jaro-Winkler string distance for names) is allowed. The matched pairs are selected based on a threshold specified by the user.

FastLink can incorporate auxiliary information such as up-weighting or down-weighting a specific value of name given its frequency. It appears to assign a match probability to each matched pair. It also offers the function to deduplicate multiple records from the same person.

No tools are provided to simulate data for linkage purposes. There are no functions to evaluate the fit of a linkage model. Impute matched sets cannot be generated.

R: Reclin

R: Reclin (https://cran.r-project.org/web/packages/reclin/index.html) is a free R package. The latest version is 0.1.2 released in November 2021. Users need to utilize R to import and read data. R programming is needed to clean and standardize variable because the package does not have built-in validation and standardization routines.

Matching variables can be matched exactly. Or a specified tolerance can be set for matching variables (e.g., Jaro-Winkler string distance for names). The matched pairs can be selected based on the highest matching score from the probabilistic linkage or a user defined threshold.

Reclin appears to assign a match probability to each matched pair. It also facilitates users to deduplicate multiple records from the same person.

There are no built-in tools to simulate data to help users develop, improve, and validate linkage models. Impute matched sets cannot be created.

Link King

As a free software, Link King (http://the-link-king.party/) runs on SAS platform. The latest version is 9.0 released in March 2018. Up to 99,999,999 records can be used in linkage. A Graphical User Interface (GUI) interface is used, so users do not need to have extensive SAS experience. The linkage algorithms were developed from linking the SubstanceAbuse and Mental Health Services Administration's (SAMHSA) Integrated Database.

Link King can read various forms of files in SAS, SPSS, comma delimited files, and Excel tables. Link King needs to use specific variables: social security number, birthdate, last name, first name, middle name, maiden name, race, sex, and client ID. To run a linkage, the following variables are required: 1) first name; 2) last name; 3) social security number or date of birth.

Exact agreement or agreement within a specified tolerance can be used to compare values from matching variables. There is a guided standardization process according to the variable types. Missing values can be defined during the standardization process. Probabilistic or deterministic linkage can be performed. Link King offers the function to deduplicate multiple records from the same person.

Simulated data cannot be generated to facilitate users to develop and refine linkage specifications. It can generate random samples of matched pairs to help users review the quality of the linkage model. Multiple imputed matched sets cannot be created.

Link Plus

As a free and standalone probabilistic linkage software, Link Plus (http://www.cdc.gov/cancer/npcr/tools/registryplus/lp.htm) was created by the Centers for Disease Control and Prevention's (CDC) Division of Cancer Prevention and Control to assist cancer registries. The latest official version is 2.0 released in June 2007.

Users can utilize Link Plus to identify duplicates in a self-match or link databases. The input data need to be fixed width text or delimited. Up to 4.8 million records can be used in linkage.

The following matching fields are permitted: first name, middle name, last name, date, social security number, character strings, and zip codes. Users can also customize variables. An exact match or fuzzy match with a tolerance can be set for matching variables. No functions are available to standardize or clean variables.

No tools are provided to simulate data or assess the quality of linkage specifications. The MCMC method is not available to refine the linkage model. Multiple imputed matched sets cannot be produced. The support for Link Plus has been depreciated by the CDC.

The following Table 1 summarizes the capabilities and Table 2 provides the strengths and weaknesses regarding the reviewed software.

Table 1. Capabilities for Selected Linkage Software							
	LinkSolv	Match*Pro	R: RecordLinkage	R: FastLink	R: Reclin	Link King	Link Plus
Cost	$3-5K	Free	Free	Free	Free	Free	Free
Platform	Access	Standalone	R	R	R	SAS	Standalone
Standardization/Data Cleaning	Yes	Yes	No	Limited	No	Yes	No
Probabilistic Match	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Deterministic Match	No	No	No	No	No	Yes	Yes
Customizable Match Weights	No	Yes	No	Yes	No	No	Yes
Custom VariableTypes	Yes	Yes	Yes	Yes	Yes	No	Yes
Fuzzy Matching Comparisons	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Model EvaluationTools	Yes	Yes	No	No	No	Limited	No
MCMC and Imputation of Missing Links	Yes	No	No	No	No	No	No
Deduplication/ SelfMatch	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 2. Strengths and Weaknesses for Selected Linkage Software		
	Strengths	**Weaknesses**
LinkSolv	• Markov Chain Monte Carlo method to impute multiple sets of matched pairs to address the limitation such as under-estimated standard errors and selection bias when using a single set of matched pairs • Assign match probabilities to match pairs • Many built-in standardization routines including deriving variables from external cause of morbidity codes • Matching can be exact or within a tolerance • Ability to simulate data to build linkage models • Ability to examine dependency among matching variables and apply penalty for two dependent variables. • Upweight or down weight a specific value of a matching variable given its frequency	• Added complexity for multiple imputation analysis • Not free
Match*Pro	• Built-in routines to validate variables including date, name, and address • Flexible in defining blocking variables • Matching can be exact or within a tolerance • Assign match probabilities to match pairs • Free	• No routine to standardize and derive variables from external cause of morbidity codes • A single set of matched pairs may be susceptible to the under-estimated standard errors and selection bias
R: RecordLinkage	• Matching can be exact or within a tolerance • An exploratory feature to apply machine learning methods to data linkage • Free	• No built-in routines to clean and standardize variables. • No assignment of match probabilities to matched pairs • A single set of matched pairs may be susceptible to the under-estimated standard errors and selection bias

R: FastLink	• Some built-in routines to standardize name and address • Flexible in defining blocking variables • Matching can be exact or within a tolerance • Use auxiliary information to upweight or down weight a specific value of a variable given its frequency • Assign match probabilities to match pairs • Free	• A single set of matched pairs may be susceptible to the under-estimated standard errors and selection bias
R: RecLin	• Matching can be exact or within a tolerance • Assign match probabilities to match pairs • Free	• No built-in routines to clean and standardize variables. • A single set of matched pairs may be susceptible to the under-estimated standard errors and selection bias
Link King	• Matching can be exact or within a tolerance • Some built-in standardization routines • Free	• Specific variables are required: 1) first name; 2) last name; 3) date of birth or social security number • A single set of matched pairs may be susceptible to the under-estimated standard errors and selection bias
Link Plus	• Matching can be exact or within a tolerance • Free	• No built-in routines to clean and standardize variables. • A single set of matched pairs may be susceptible to the under-estimated standard errors and selection bias • The support for Link Plus has been depreciated.

References

1. National Center for Injury Prevention and Control. Linking Information for Nonfatal Crash Surveillance (LINCS): A guide for integrating motor vehicle crash data to help keep Americans safe on the road. 2019. https://www.cdc.gov/transportationsafety/linkage/Linking-Information-Nonfatal-Crash-Surveillance.html. Accessed 12/21/2021.
2. Cook LJ, Thomas A, Olson C, Funai T, Simmons T. *Crash Outcome Data Evaluation System (CODES): an examination of methodologies and multi-state traffic safety applications.* 2015.