Frederick National Laboratory
for Cancer Research

sponsored by the National Cancer Institute

EVALUATING THE PERFORMANCE OF PRIVACY PRESERVING RECORD LINKAGE SYSTEMS (PPRLS)

Evaluation performed by Information Management Services (IMS) for Leidos Biomedical Research (LBR) under the agreement 20Q035TO01, issued as a subcontract under Contract HHSN2612015000031, Task Order No. HHSN26100038 issued by the National Cancer Institute (NCI), National Institutes of Health (NIH), Department of Health and Human Services (HHS)

Submitted on March 27, 2023

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Privacy Preserving Record Linkage Systems (PPRLS) mask sensitive information about the entities they are linking through a process such as encryption and secure multi-party computation or hashing (also sometimes referred to as tokenization or deidentification). During the hashing process, a series of cryptographic functions are applied to the input data—consisting of Personally Identifiable Information (PII)—to generate one-way hashes. After the input data have been hashed/tokenized, the linkage process compares the hash tokens across the records in two or more files to identify sets of records that are believed to belong to the same person.

The proficiency at which PPRLS accurately link individuals across different data sources—both to link disparate data for more comprehensive information on an individual and for longitudinal analyses—is a topic of interest to the National Cancer Institute (NCI). For the results of a longitudinal analysis to be valid, the correct information needs to be linked to each patient; therefore, the PPRLS should minimize the number of false positives and false negatives. The aim of this evaluation was to assess the speed and accuracy of several commercially available PPRLS selected from a prior landscape analysis [1], using data for which the PII represented a wide range of data quality and completeness.

We obtained source data for this assessment from linkages that were previously conducted using raw (non-deidentified) PII from six U.S. central cancer registries, CVS, and Walgreens. The linkages performed for this evaluation aimed to represent three real-world data scenarios. In Scenario 1, the PII was both highly accurate (very few errors) and complete (very few missing values). In Scenario 2, the data were slightly less accurate but Social Security number (SSN), which is perhaps the single most effective identifier, was missing on the vast majority of records. In Scenario 3, the PII was altered by introducing spelling errors, transposed numbers, and so on to evaluate how the PPRLS performed in situations where the input data were highly imperfect.

For each scenario, the source data were linked in a probabilistic, non-privacy preserving record linkage system. The results from this linkage were manually reviewed to create a “gold standard” set of confirmed person-level matches.

The PII from each data source was subjected to the encryption function of each PPRL system to create de-identified hashed tokens that were subsequently linked using the PPRLS. The matches obtained from the PPRLS were compared against the gold standard matches, and the numbers of true positives, false positives, true negatives, and false negatives were determined. Those numbers were used to derive several quantitative metrics to assess the quality of the results obtained from each PPRLS. These quantitative metrics included precision, recall, the F1 score (the harmonic mean of precision and recall), the false discovery rate, accuracy, and specificity.

This report summarizes evaluation results of PPRLS from four vendors (Datavant, HealthVerity, Senzing, and Veeva). The F1 scores of the systems evaluated for Scenario 1 (highly accurate PII) ranged from 0.574 - 0.964. The F1 scores for Scenario 2 (missing SSN) ranged from 0.123-0.964. The F1 scores for Scenario 3 (less accurate or missing PII) ranged from 0.052-0.786. The evaluation attempted to present a fair and unbiased comparison between different PPRLS and to understand the situations in which different PPRLS might be most useful. The strengths and weaknesses of each PPRLS, are presented in the “Discussion” section.

INTRODUCTION

This project has been funded with Federal funds from the National Cancer Institute, National Institutes of Health, under Contract No. HHSN261201500003I, Task Order No. HHSN26l00038.

The NCI convened the Blue-Ribbon Panel (BRP) in 2016 to recommend areas of scientific opportunity for the NCI to pursue, which are aligned with the mission of the Beau Biden Cancer Moonshot initiative. One of the 10 areas identified by the BRP is the creation of a National Cancer Data Ecosystem (NCDE) to enable basic science researchers, clinicians, and patients to contribute, share, combine, and analyze cancer relevant data. Critical to the success of the NCDE will be its ability to integrate data from a variety of sources to increase the depth and breadth of information available to cancer researchers, without compromising patient privacy or confidentiality.

Linking data at the patient level poses several challenges, including protecting patient confidentiality, maximizing data security, achieving scalability, ensuring accuracy in linking the same patient, and assuring consistency of PII across a variety of sources that may hold different components of PII (*e.g.*, name and date of birth [DOB] versus SSN and DOB). To address this, a Task Order (TO) was funded under the Cancer Moonshot, Cancer Data Ecosystem to identify and evaluate existing software tools that enable privacy-preserving linkages of different data sources. This type of linkage is performed through hash tokens, which are used to link individual patient data from disparate sources without releasing or sharing PII.

The TO was executed in two phases. Phase one consisted of the detailed landscape analysis of existing encryption software systems and tools (PPRLS). The landscape analysis began in January 2019. After its completion in February 2020, the four software products that scored the highest in the composite assessment portion of the analysis were selected for evaluation in Phase two: Datavant, HealthVerity, Senzing, and Veeva.

The objective of the Phase two evaluation is to independently compare the performance of each of the four selected PPRLS to the same set of "gold standard" linkages. The "gold standard" linkages are based on identifiable PII with human adjudication using data representing three different scenarios of real-world PII data quality and completeness. The Phase two evaluation for the HealthVerity and Datavant software began in July 2020 and concluded in June 2021. The Phase two evaluation for Senzing and Veeva began in January 2022 and concluded in November 2022. The purpose of this report is to: 1) describe the methodology adopted for the Phase two evaluation; and 2) present findings from the evaluation of the HealthVerity, Datavant, Senzing, and Veeva software.

MATERIALS AND METHODS

Data Sources

Three linkages were conducted to assess the performance of each privacy-preserving record linkage system under a specific set of operating conditions in which the levels of PII data quality varied from high, to moderate, to low. The files used in each linkage contained the first name, middle name, last name, maiden name, sex, DOB, SSN, telephone numbers, and addresses for each patient.

Scenario 1 was designed to represent a linkage between datasets with high PII data quality (complete and accurate PII with SSN). For this component, files were obtained from six of the NCI's Surveillance, Epidemiology, and End Results (SEER) registries (Connecticut, Georgia [GA], Iowa, Louisiana, New Jersey, and New York); the files contained PII from individuals who were diagnosed with cancer in each state between 2005 and 2015. To enable linkage for cancer patients who may have moved, the data included each individual's longitudinal residential address and telephone number history. The SEER cancer registries regularly review and validate their data, resulting in consistent and accurate PII. Levels of completeness are also very high, with most of the PII fields for this linkage having completion rates of over 97%. Only 6,648 individuals out of 2,936,933 appeared in more than one registry, representing an overlap of 0.23%. The number of false positives tends to increase linearly with the size of the data sets, as the chance for coincidental matches increases. The degree of overlap can significantly impact the number of false positives even if the number of true positives remains constant, thereby changing the precision and F1 score.

Scenario 2 was designed to represent a linkage between datasets with moderate and high PII data quality (with one of the files largely missing SSN). The files used in this linkage were obtained from the GA cancer registry. The first file contained data from the GA cancer registry database. The second file contained pharmacy claims data, which were provided to the GA registry by CVS and Walgreens under public health reporting. The records from the GA cancer registry were for individuals who were diagnosed with cancer between 2000 and 2017 in the state of Georgia. The pharmacy claims from CVS and Walgreens were generated for GA residents who filled prescriptions for anti-neoplastic (chemotherapy) medications between 2013 and 2020. The pharmacy data included oral cancer treatment as required by state cancer reporting procedures and mandates and contained some typographical errors (typos), lower levels of completeness and very few SSNs. Among all the identifiers in the pharmacy dataset, SSN had the lowest completeness rate of approximately 0.6% vs. 97.5% in the GA registry data. Both the GA registry data and the pharmacy data included the longitudinal residential address and telephone number history for each individual. The ability to link via longitudinal addresses and phone numbers was an important additional feature that enhanced the ability to link over time as a confirmatory variable when SSN was not available. There were 41,488 individuals out of 97,096 that appeared in both the GA registry and CVS/Walgreens data, representing an overlap of 42.7%.

Scenario 3 was designed to test a situation in which datasets have low data quality as indicated by missing or incomplete data or errors such as typos in the patient's name or address. A SAS program was developed to modify the GA cancer registry and CVS/Walgreens data used in Scenario 2 to create the datasets used in Scenario 3. The SAS program generated random numbers between 1 and 100. The random number generator was called once for every PII value across every record in both files from Scenario 2. If the generator produced a number that was less than or equal to 15, then the data for the current field (name, DOB, SSN, address, etc.), for the current record, were flagged for modification and then altered in some way. The list of data alterations included: transposition of characters within the field, truncation of the field, introduction of spelling errors (*e.g.,* doubling or removal of some characters), complete replacement of the field's value with bad data, and complete replacement of the field's value with a missing (*i.e.,* blank) value. Each of these alterations had an independent 10 percent chance of occurring, and the program was structured in such a way as to guarantee that at least one of these alterations would be made if the original value had been flagged for modification. It was therefore possible (albeit rare) for multiple alterations to stack on top of one other. Overall, there was a 15 percent chance that any value, chosen at random, would be altered in some way. It was also possible for multiple values within each record to have been altered. The result was that the modified GA cancer registry data and CVS/Walgreens data used in Scenario 3 each had significantly lower levels of accuracy and completeness than the original datasets that were selected for use in Scenario 3.

The source datasets for each scenario were already in use as part of cancer surveillance operations. Permission was obtained from the owners of each data file described above for use in this evaluation. Information Management Services, Inc. (IMS) performed the evaluation. IMS established Interconnection and Security Agreements (ISAs) with the registries long before the initiation of this evaluation and serves as the registries' trusted third party and honest broker. In this role, IMS maintains each registry's database (which contains PII) and performs centralized linkages on behalf of registries. IMS complied with Federal Information Security Modernization Act (FISMA) standards when handling the data for this project.

Evaluation Workflow

We adopted the following workflow to evaluate the performance of each selected PPRLS using data under each respective scenario (See *Figure 1*):

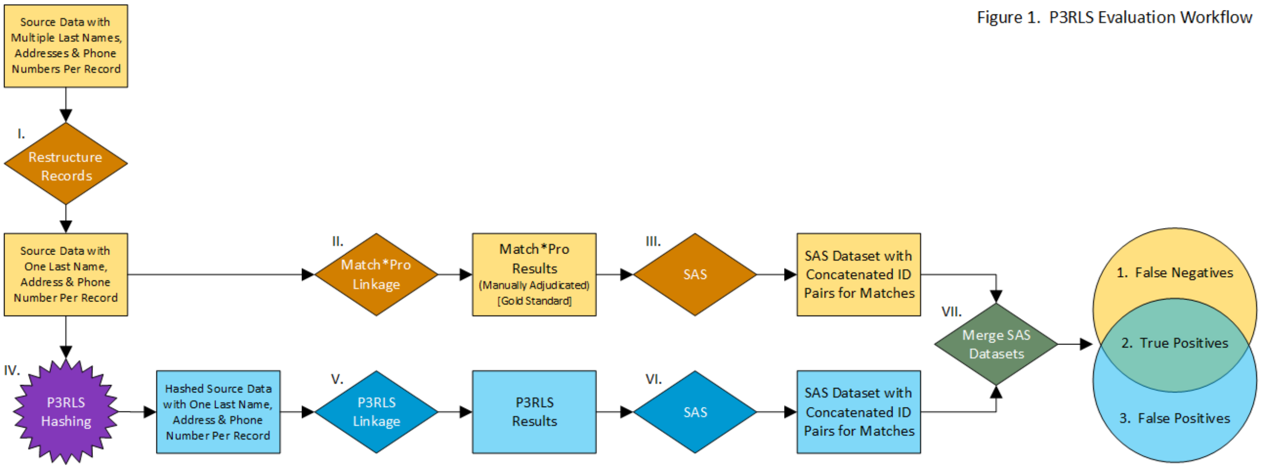
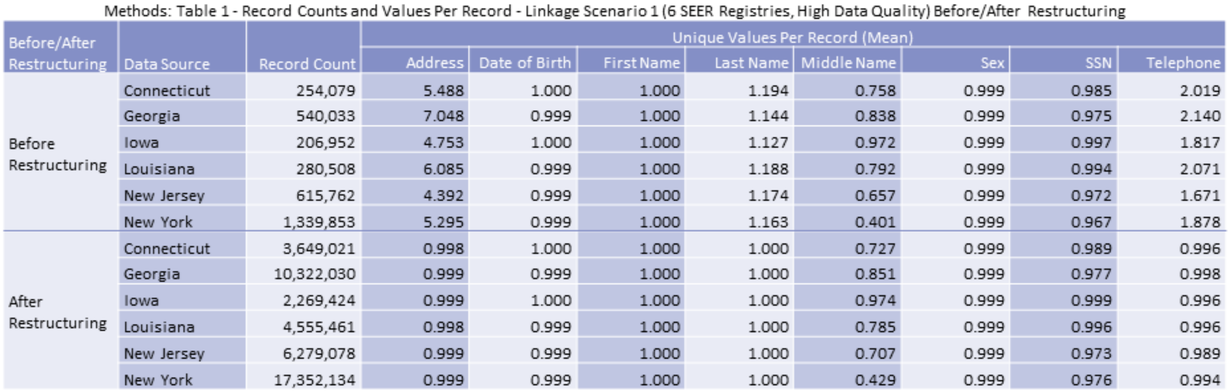

Source Data with Multiple Last Names, Addresses & Phone Numbers Per Record
I.
Restructure Records
Source Data with One Last Name, Address & Phone Number Per Record
II.
Match*Pro Linkage
Match*Pro Results (Manually Adjudicated) [Gold Standard]
III.
SAS
SAS Dataset with Concatenated ID Pairs for Matches
IV.
P3RLS Hashing
Hashed Source Data with One Last Name, Address & Phone Number Per Record
V.
P3RLS Linkage
P3RLS Results
VI.
SAS
SAS Dataset with Concatenated ID Pairs for Matches
VII.
Merge SAS Datasets
1. False Negatives
2. True Positives
3. False Positives

Figure 1. P3RLS Evaluation Workflow

I. **<u>Restructure the Source Data</u>**

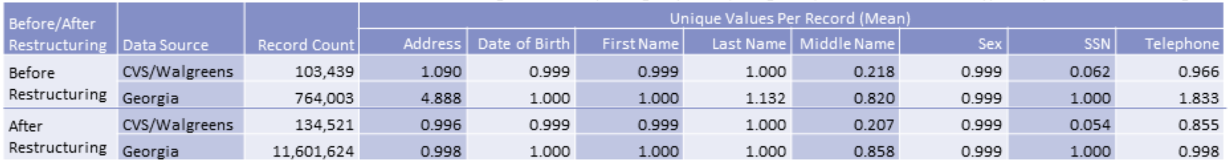
Under each scenario, the source data (two input files) included multiple last names, addresses, and phone numbers on each record (with some records having as many as 30 addresses and phone numbers), but two of the PPRLS software we evaluated (Datavant and HealthVerity) could only accept a single value per identifier per record. To address this limitation, the source data were restructured so that a single last name, address, and phone number were present on each record, with each combination of last name, address, and phone number in the original dataset resulting in the creation of a new record in the restructured dataset (*e.g., a record containing one last name, two addresses, and three phone numbers in the original dataset would be split into six records in the restructured dataset*). The number of records in each dataset and the mean number of values for each of the PII elements used to link the data, both before and after the records were restructured, can be found in *Methods: Tables 1-3*.

Methods: Table 1 - Record Counts and Values Per Record - Linkage Scenario 1 (6 SEER Registries, High Data Quality) Before/After Restructuring



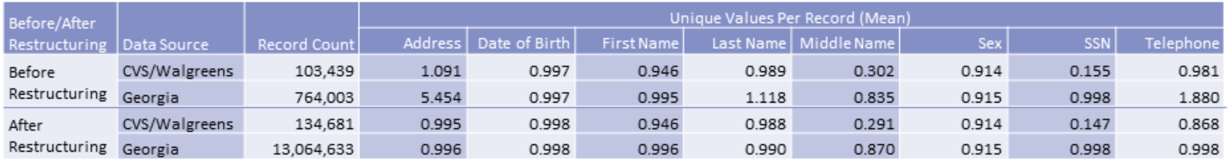
Before/After Restructuring	Data Source	Record Count	Unique Values Per Record (Mean)							
			Address	Date of Birth	First Name	Last Name	Middle Name	Sex	SSN	Telephone
Before Restructuring	Connecticut	254,079	5.488	1.000	1.000	1.194	0.758	0.999	0.985	2.019
	Georgia	540,033	7.048	0.999	1.000	1.144	0.838	0.999	0.975	2.140
	Iowa	206,952	4.753	1.000	1.000	1.127	0.972	0.999	0.997	1.817
	Louisiana	280,508	6.085	0.999	1.000	1.188	0.792	0.999	0.994	2.071
	New Jersey	615,762	4.392	0.999	1.000	1.174	0.657	0.999	0.972	1.671
	New York	1,339,853	5.295	0.999	1.000	1.163	0.401	0.999	0.967	1.878
After Restructuring	Connecticut	3,649,021	0.998	1.000	1.000	1.000	0.727	0.999	0.989	0.996
	Georgia	10,322,030	0.999	0.999	1.000	1.000	0.851	0.999	0.977	0.998
	Iowa	2,269,424	0.999	1.000	1.000	1.000	0.974	0.999	0.999	0.996
	Louisiana	4,555,461	0.998	0.999	1.000	1.000	0.785	0.999	0.996	0.996
	New Jersey	6,279,078	0.999	0.999	1.000	1.000	0.707	0.999	0.973	0.989
	New York	17,352,134	0.999	0.999	1.000	1.000	0.429	0.999	0.976	0.994

Methods: Table 2 - Record Counts and Values Per Record - Linkage Scenario 2 (GA Registry & CVS/Walgreens, Moderate Data Quality) Before/After Restructuring



Before/After Restructuring	Data Source	Record Count	Unique Values Per Record (Mean)							
			Address	Date of Birth	First Name	Last Name	Middle Name	Sex	SSN	Telephone
Before Restructuring	CVS/Walgreens	103,439	1.090	0.999	0.999	1.000	0.218	0.999	0.062	0.966
	Georgia	764,003	4.888	1.000	1.000	1.132	0.820	0.999	1.000	1.833
After Restructuring	CVS/Walgreens	134,521	0.996	0.999	0.999	1.000	0.207	0.999	0.054	0.855
	Georgia	11,601,624	0.998	1.000	1.000	1.000	0.858	0.999	1.000	0.998

Methods: Table 3 - Record Counts and Values Per Record - Linkage Scenario 3 (GA Registry & CVS/Walgreens, Modified PII, Low Data Quality) Before/After Restructuring



Before/After Restructuring	Data Source	Record Count	Unique Values Per Record (Mean)							
			Address	Date of Birth	First Name	Last Name	Middle Name	Sex	SSN	Telephone
Before Restructuring	CVS/Walgreens	103,439	1.091	0.997	0.946	0.989	0.302	0.914	0.155	0.981
	Georgia	764,003	5.454	0.997	0.995	1.118	0.835	0.915	0.998	1.880
After Restructuring	CVS/Walgreens	134,681	0.995	0.998	0.946	0.988	0.291	0.914	0.147	0.868
	Georgia	13,064,633	0.996	0.998	0.996	0.990	0.870	0.915	0.998	0.998

Note: After restructuring, the mean number of values shown is the same as the completion rate.

II. Link the Restructured Source Data to Create a "Gold Standard" Dataset

Once the two input datasets were restructured, a probabilistic record linkage application called Match*Pro was used to link the data. The record pairs generated by Match*Pro were manually adjudicated and classified as being either a match or a non-match. Record pairs that were questionable were further adjudicated by the cancer registries holding the cases prior to classification. Record pairs that were classified as matches, following manual adjudication, were stored in the "gold standard" SAS dataset for our analysis. The number of **true negatives (TN)** was obtained by counting the number of individuals who did not have any matches.

III. Create SAS Datasets from the Match*Pro Results

SAS programs were developed to read the gold standard pairs from the Match*Pro linkage results to construct SAS datasets. The IDs from the two records in each linked pair were concatenated to create a composite ID (*e.g.*, a patient from file 1 with the ID "ABC" is linked to a patient in file 2 with the ID "123" to produce the composite ID "ABC-123"). The composite ID served as the primary key for our evaluation.

IV. Hash the Source Data

The restructured source data were hashed/de-identified using each PPRLS vendor's tokenization/deidentification software.

V. Link the Hashed Source Data using PPRLS

The de-identified source data were linked using each vendor's PPRLS matching software.

VI. Create SAS Datasets from the PPRLS Results

Upon completion of the linkage, each of the PPRLS matching applications produced a text file containing a list of linked IDs. SAS programs were developed to read these lists and transform them into SAS datasets. During this process, the ID pairs from the lists were concatenated to create a composite ID, which served as the primary key for our evaluation.

VII. Merge the SAS Datasets

The SAS datasets created from the PPRLS results were merged with the gold standard results (*i.e.,* the SAS datasets created from the Match*Pro gold standard results) using the composite IDs in each of the datasets. The process of merging the data identified three sets of ID pairs for each software system:

1. A set of **false negative (FN)** pairs, which consisted of ID pairs from the Match*Pro gold standard results that the PPRLS software failed to identify.

2. A set of **true positive (TP)** pairs, which included ID pairs from the Match*Pro gold standard results that the PPRLS software also identified.

3. A set of **false positive (FP)** pairs, which contained ID pairs that were identified by the PPRLS software that were not included in the Match*Pro gold standard results.

The false negatives were manually reviewed to identify trends in the data that could be used to explain why the PPRLS software failed to identify a match; for example, the same address might have been formatted differently in each file (*e.g.*, 123 Fifth Avenue vs. 123 5th Ave). Likewise, the false positives were manually reviewed to identify trends in the data that could be used to explain why the PPRLS software erroneously identified a match and to ensure that our gold standard results were complete. The analysis of FPs is particularly important for many end users as linking individuals and incorrectly producing combined data may result in incorrect conclusions of an analysis, and thus is typically a more important measure than FNs or missing true positives.

Quantitative Metrics

Once the numbers of true positives, false positives, and false negatives were obtained, several quantitative metrics were derived to assess the quality of the PPRLS linkage results, including:

1. **Precision (Positive Predictive Value)** =TP / (TP + FP) = the proportion of returned matched pairs that were true matches

2. **Recall (Sensitivity)** = TP / (TP + FN) = the proportion of all true matched pairs that were correctly identified

3. **F-Score** = (2 x Precision x Recall) / (Precision + Recall) = the harmonic mean of Precision and Recall

4. **False Discovery Rate (FDR)** = FP / (FP + TP) = the proportion of returned matched pairs that were false matches

5. **Accuracy** = (TP + TN) / (TP + TN + FP + FN) = the proportion of all pairs that were classified correctly as matches or non-matches

6. **Specificity (True Negative Rate)** = TN / (TN + FP) = the proportion of all true non-matched pairs that were correctly identified

Performance and Scalability Benchmarks

The following metrics were used to further assess the performance and scalability of each software:

1. **Runtime**: how long an application took to perform a task

2. **CPU Usage**: the number of CPU cores that the application used expressed as a percentage of the total number of CPU cores that were available

3. **Memory Usage**: the amount of memory (RAM) that the application used

4. **Network Communication Size**: the amount of data that the application passed through a communications network within the IT infrastructure.

The performance and scalability benchmarks for Datavant and HealthVerity were obtained using a desktop computer running in a Microsoft Windows 10 Enterprise environment. The computer had an Intel i3-8100 CPU (a 64-bit quad-core processor that operates at 3.60GHz) and 64GB of DDR3 memory installed.

The performance and scalability benchmarks for Veeva and Senzing were obtained using a Linux server environment that was configured to each vendor's specifications. Additional information regarding these specifications can be found in the *Deployment – Senzing* and *Deployment – Veeva SafeMine* sections below.

PPRLS Software Introduction

In this report, four commercial vendors' PPRLS were evaluated: Datavant, HealthVerity, Senzing, and Veeva. Details of the tokenization and matching processes used by each vendor are provided below.

HealthVerity Tokenization

HealthVerity's Census technology supports and deploys privacy preserving record linkage at the patient level. Each individual in the output of the de-identification engine is represented by a set of corresponding tokens generated as a byproduct of their input Protected Health Information (PHI). The output tokens are represented in the form of salted, one-way cryptographic hashes. These hashes are implemented using a standard Secure Hashing Algorithm (SHA-256), and as noted, common salts are inserted into the hashing as part of the automated build process by a third party. This is sufficient to represent many PHI elements, such as DOB or internal patient identifiers. However, achieving accurate record linkage using such hashes is non-trivial because a patient's record often contains typographical and semantic errors.

To overcome this problem, for fields that are likely to contain typographical errors and are not directly attackable using a constraint satisfaction program, HealthVerity will split a patient's identifier into a series of *n-grams*, whereby the patient's identifiers are split into a series of overlapping substrings that are subject to a set of hash functions and mapped into a Bloom Filter (BF), which is a binary array of a fixed length. The use of BFs produces a significant improvement in matching accuracy, as traditional means of hashing are fully deterministic and, as a result, are susceptible to significant false negative errors in matching due to source data variance.

healthverity™

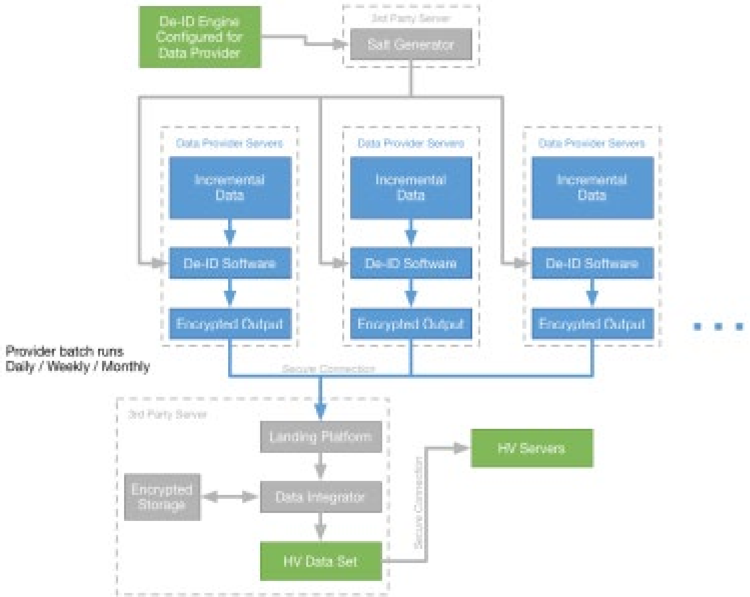


Figure 2. General architecture for RBF generation.

The image shown above originated from one of HealthVerity's whitepapers and was reprinted here with HealthVerity's permission.

The de-identification software is distributed as a Java.jar file and is executed from the command line. HealthVerity provides its clients with customized versions of the de-identification engine that are designed to conform with the client's input record layout; therefore, the version of the de-identification engine that was provided to the NCI and IMS for evaluation purposes was tailor-made to process the restructured source data.

The de-identification engine produces an encrypted file. This means that the tokens are not visible when the file is viewed through a text editor.

One feature that differentiates HealthVerity from some other PPRLS vendors is the fact that the matching takes place exclusively at HealthVerity, rather than on the client's PC or server. This allows the data to be matched against a much larger encrypted and tokenized dataset that is maintained by HealthVerity. This opportunity to match against a larger file potentially provides additional opportunity to match, leveraging a broader set of PII data. HealthVerity has set up a secure FTP where clients can log in and upload the encrypted files that are produced by the de-identification engine. After each of the input datasets were de-identified, each of the encrypted files were uploaded to HealthVerity's servers using the FTP.

HealthVerity Matching

HealthVerity's matching engine uses a probabilistic process capable of addressing noisy data common to patient identifiers, including typographical errors, nicknames, missing data, and changing information. Each candidate match is evaluated based upon the evidence present in the observed tokens. A probability is assigned to each token, based on how well the tokens match. Even missing information is assigned a probability. In the case of BF tokens, a bit difference between the queried token and the candidate match can reveal the likelihood of a small change consistent with a typographical error. The only output of the matching is a common HealthVerity ID (HVID), a longitudinal patient identifier regardless of space, dataset, or time.

As previously stated, all matching is performed behind HealthVerity's firewall on *their* servers. Once we received notification from HealthVerity that the matching was completed, we logged into the secure FTP to retrieve our results, which were in a pipe-delimited file format.

HealthVerity PPRLS Performance Tuning

Due to the flexible nature of HealthVerity's Census technology, the de-identification engine can be tuned by use case to target requested false positive and false negative rates. To support the NCI for this evaluation, HealthVerity defined and adjusted the de-identification engine's false positive and false negative ratios to provide a target 0.01 false positive rate and defined patient confidence levels.

To facilitate this level of specification, HealthVerity first processed the SEER registry data through the standard de-identification engine to populate an initial list of HVIDs. Once these de-identified patient identifiers were populated per patient, candidate matches across state registries were determined. Upon confirmation of a candidate match between registries, HealthVerity evaluated each patient match by risk of its unique potential to be a false positive match. This risk evaluation was completed using the inherent ambiguity of the tokenized data with regard to name, DOB, location, and SSN, potential variations based on common errors (*e.g.,* typos) or changing data (*e.g.,* patterns of change of residence), as well as consideration of field population density.

Taken together, these variable evaluations and determined densities form a local distance measure, like a Mahalanobis distance, to identify the relative likelihood that a given candidate match was a false positive, based on coincidental similarity of the tokens. This distance measure uses a threshold to identify the high-confidence matches that are expected to maintain the desired low false positive rate, as well as the low-confidence matches that could lead to a higher false positive rate.

We evaluated the performance of the standard matching approach and the tuned matching approach when linking the SEER registry data. We have presented both outcomes in our Results section, which follows.

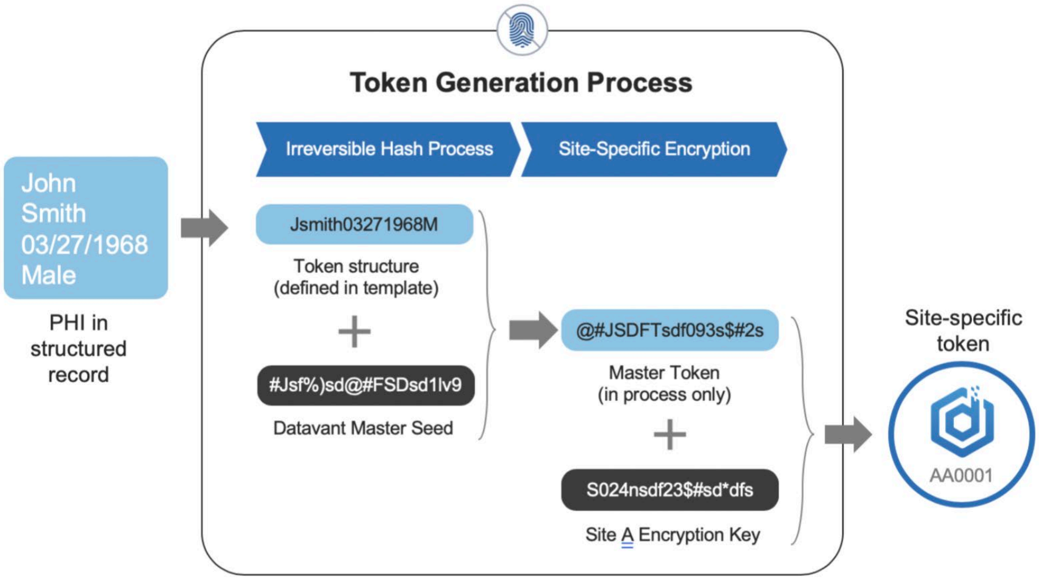
Datavant On-Premises Tokenization

Datavant's on-premises version of the tokenization software, **DeID** version 3.8.1, was used in this evaluation. The software applies configurable de-identification rules to an input flat file of data and generates Datavant tokens. DeID performs automated data cleaning and standardization operations prior to tokenization, including stripping leading and trailing whitespace, reformatting dates into "YYYYMMDD" format, and capitalizing all alphabetic characters.

DeID applies validation rules by input field. For example, the SSN field must be a valid U.S. Social Security Number in format FNN-NN-NNNN or FNNNNNNNN, where F is a numeric digit from 0-8, and N is a numeric digit from 0-9. The first three digits cannot be 666 and none of the three sections (first three, middle two, or last four digits) can be all zeroes. If a given input field does not meet validation rules, DeID generates an error code in place of a Datavant token. There are certain pre-cleaning processes that Datavant recommends end-users undertake prior to running the Datavant software.

Token generation is a two-step process:

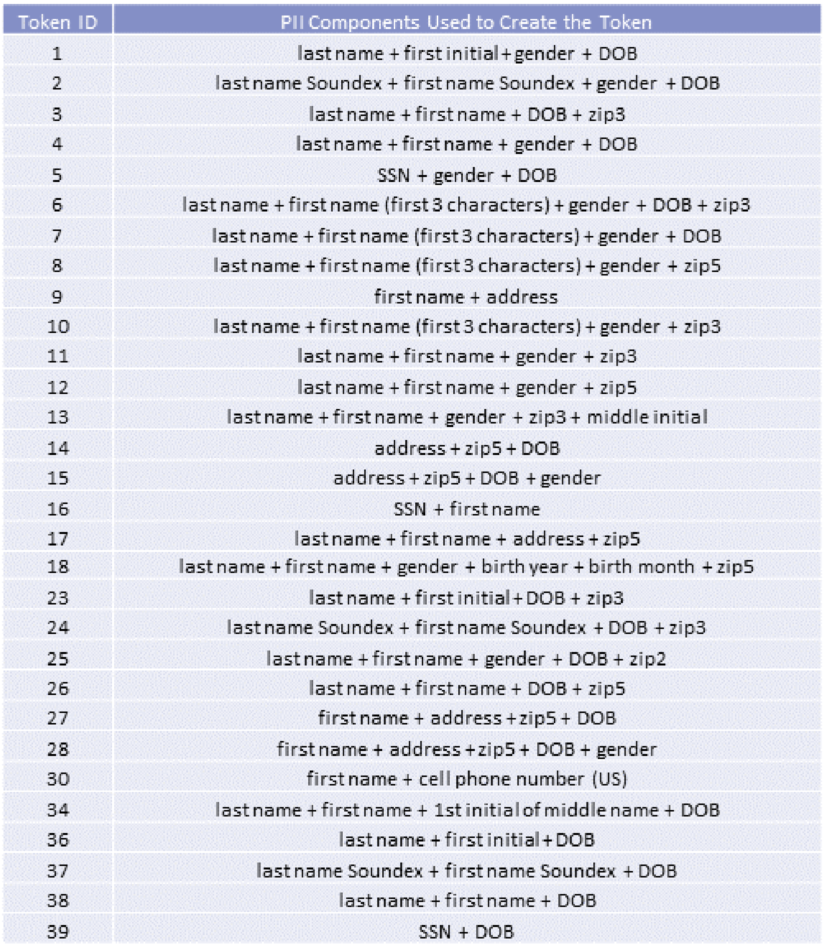
1. SHA-256 Hash with Master Salt - makes tokens irreversible
2. Encryption - makes tokens site-specific to protect against security breaches



The image shown above originated from one of Datavant's whitepapers and was reprinted here with Datavant's permission.

DeID produces a comma-delimited output file that contains the original (*i.e.,* untokenized) patient ID followed by multiple hash tokens across several columns. Each token is created using a different combination of PII elements from the source data. The number and types of tokens produced by the software are configurable both by Datavant and by the user. Datavant created a set of configurations designed to accommodate the restructured versions of the source data and to produce a set of tokens based on the information available in the files. For a more detailed description of each of the tokens that were produced, see *Methods: Table 4*.

Methods Table 4 – Datavant Tokens Created



Token ID	PII Components Used to Create the Token
1	last name + first initial + gender + DOB
2	last name Soundex + first name Soundex + gender + DOB
3	last name + first name + DOB + zip3
4	last name + first name + gender + DOB
5	SSN + gender + DOB
6	last name + first name (first 3 characters) + gender + DOB + zip3
7	last name + first name (first 3 characters) + gender + DOB
8	last name + first name (first 3 characters) + gender + zip5
9	first name + address
10	last name + first name (first 3 characters) + gender + zip3
11	last name + first name + gender + zip3
12	last name + first name + gender + zip5
13	last name + first name + gender + zip3 + middle initial
14	address + zip5 + DOB
15	address + zip5 + DOB + gender
16	SSN + first name
17	last name + first name + address + zip5
18	last name + first name + gender + birth year + birth month + zip5
23	last name + first initial + DOB + zip3
24	last name Soundex + first name Soundex + DOB + zip3
25	last name + first name + gender + DOB + zip2
26	last name + first name + DOB + zip5
27	first name + address + zip5 + DOB
28	first name + address + zip5 + DOB + gender
30	first name + cell phone number (US)
34	last name + first name + 1st initial of middle name + DOB
36	last name + first initial + DOB
37	last name Soundex + first name Soundex + DOB
38	last name + first name + DOB
39	SSN + DOB

The DeID software is available as a command-line executable and as a Windows application (with a graphical user interface / GUI), but there are limits placed on the number of records that can be processed efficiently with the Windows application version. The work conducted for this evaluation was performed using the command-line version of the application.

Datavant On-Premises Matching

Datavant tokens are non-proprietary. The end-user can formulate their own matching algorithms on the tokens. Since the token design and form are established and disclosed to the end-user, the end-user can apply different matching algorithms for their linkage use cases.

Matching is a two-step process:

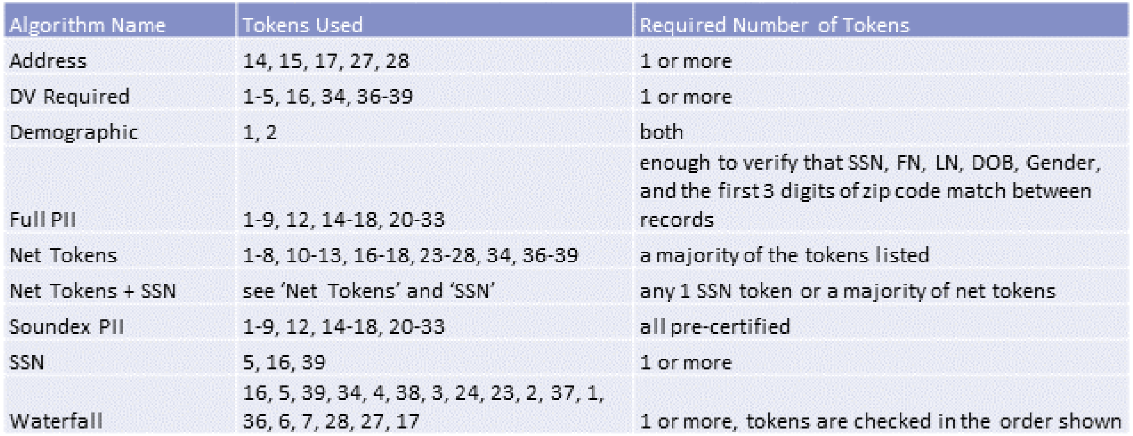
1. Enable token interoperability – the tokens for each authorized data recipient are converted/translated into a shared encryption scheme.
2. Apply matching algorithm - the records are linked by running the matching algorithms on the translated tokens.

Datavant's **Link** software translates between encryption keys to allow data partners to share and match data based on their data governance framework. For data source A to share tokens with data recipient B, both sites tokenize their data into their own site-specific encryption keys. Site A runs Link-Send to translate their tokens into a shared transit scheme with site B, and those transit tokens will be delivered to site B. Site B runs Link-Receive to translate the transit tokens into site B encrypted tokens, allowing them to perform matching with their own de-identified data in the same scheme. The purpose of this process is to ensure that tokens associated with a specific initiative can only be processed by the authorized party. Likewise, if a data partner sends tokens intended for a different initiative to their authorized processor, that party will be unable to link and match on the non-interoperable tokens.

Datavant's Link software is available as a command-line executable and as a Windows application. There are limits placed on the number of records that can be processed efficiently with the Windows application version. These limits did not apply to the command-line version of the application, so we used version 3.8.1 of the command-line executable for this evaluation.

Datavant's **Match** software assigns a match ID to each patient based on a comparison between the tokens for that patient and the tokens generated for other patients. Patients that are assigned the same match ID are called a match. Nine "standardized" matching algorithms are available for users of the on-premises Match software. These algorithms were derived from the most common algorithms Datavant customers use and have tested. Each algorithm looks at a different subset of the tokens that were produced, and each applies a different comparison logic to identify a match. The different algorithms are intended to be used in reference to specific sets of selected tokens, thus allowing fit-for-purpose matching options. All nine of the algorithms were tested in this evaluation. For a description of each matching algorithm, see *Methods: Table 5*.

Methods Table 5 - Description of Datavant Matching Algorithms



Algorithm Name	Tokens Used	Required Number of Tokens
Address	14, 15, 17, 27, 28	1 or more
DV Required	1-5, 16, 34, 36-39	1 or more
Demographic	1, 2	both
Full PII	1-9, 12, 14-18, 20-33	enough to verify that SSN, FN, LN, DOB, Gender, and the first 3 digits of zip code match between records
Net Tokens	1-8, 10-13, 16-18, 23-28, 34, 36-39	a majority of the tokens listed
Net Tokens + SSN	see 'Net Tokens' and 'SSN'	any 1 SSN token or a majority of net tokens
Soundex PII	1-9, 12, 14-18, 20-33	all pre-certified
SSN	5, 16, 39	1 or more
Waterfall	16, 5, 39, 34, 4, 38, 3, 24, 23, 2, 37, 1, 36, 6, 7, 28, 27, 17	1 or more, tokens are checked in the order shown

For a description of the tokens see: Methods Table 4

At the time of this evaluation, which began in July 2020, Datavant's Match software existed exclusively as a command-line executable and produced comma-delimited output files. The on-premises matching application that was assessed in this evaluation (version 3.7.1) was offered to partners that have data governance frameworks that only permit the designated party to hold tokens and data.

Senzing Selective Field Hashing

Senzing's method of multi-party matching using one-way hashes, whereby any matching records can be selectively reidentified if all parties agree, is called "Selective Field Hashing". The general process that can be followed to ensure discovery without disclosure would be:

1. The parties that need to share data begin by cryptographically hashing their data's identity attributes, e.g., names, addresses, IDs, phones, etc. with a shared Secret Key (or "Salt").
2. The hashed data is sent to a neutral (ideally trusted) third party who can rehash the hashed data again for extra protection (using a different "Secret Key").

3. The third party uses Senzing software on the double-hashed data to identify who is who and who is related to who.
4. Pointers to specific records sharing a nexus are revealed by the third party, (*e.g.,* party one will be informed that patient #21 in their dataset matches patient #47 in party two's dataset).
5. Both groups check those files and unilaterally decide if it is legal, and within policy, to have a more revealing discussion. If not, nothing more is learned.

Senzing implements a Public-Key Cryptography Standard (PKCS)#11-interfaced secure store that leverages AES-128 encryption using a PKCS#5 like password expansion technique augmented with a SHA2-256 hash.

Secret keys are automatically generated (per instance) using OpenSSL's random bytes generator. The secret key is then placed directly into the secure store (never seen by human eyes). Alternatively, if a user's policies require it, a user can provide a secret key and store it in the secure store. When hashing utilities are deployed to multiple locations, the secret key is shared by copying the secure store.

By default, Senzing uses an HMAC-SHA2-256 one-way hashing algorithm with a 1024-bit secret key. While this construct was developed for use in Internet Protocol Security (IPSec), it is used here for entity resolution with no modifications.

The set of fields that are hashed varies by the implementation. Attribution fields (source system and record ID) are typically not hashed.

Where the fields are hashed is implementation-specific, often following one of these patterns:

1. Local Hashing: As records are being ingested into the Senzing ER engine, selected values are hashed as described above. The clear text is discarded. The hashed values are retained in the database.
2. Single-tier Hashing: Before records are transmitted from a source system to a secondary system, selected values are hashed as described above. The recipient system then performs entity resolution using the hashed values. The hashed values are retained in the database.
3. Multi-tier Hashing: Before records are transmitted from a source system to a secondary system, selected values are hashed as described above. The independent recipient re-hashes using a different secret key. "n" re-hashing tiers can be implemented. Entity resolution is performed on the final hashed values. These hashed values are retained in the database.

Senzing is a Linux shared library with Python and Java interfaces. The software can run on any supported Linux environment, in cloud or on-premises, as long as it is co-located to a supported relational database. We conducted this evaluation using an on-premises deployment of Senzing version 2.2.

When the selective hashing process is finished, a results file containing the hash tokens is produced. This file is written in the JSON file format.

For additional information and implementation best practices please refer to this document: https://senzing.zendesk.com/hc/en-us/articles/360000970834-Selective-Field-Hashing

Senzing Principle-Based Entity Resolution

Senzing software uses a unique principle-based approach to entity resolution that eliminates the need for pre-training, tuning, or experts. Principles are based on the expected behaviors of entity attributes, e.g., names, addresses, and identifiers. For example, SSNs typically point to only one person, but DOBs behave differently, as many people share the same DOB. There are always exceptions. These exceptions are learned in real-time, as new data is received. For example, when multiple people are using the same SSN, the software detects it, labels that SSN as generic and reevaluates all prior records with that number.

Senzing's principle-based method assigns three behaviors to each entity attribute:

- **Frequency** – Do one, few, many, or very many entities generally share the same value? For example, SSN is commonly used by one entity, an address is shared by a few, and a DOB is shared by many.
- **Exclusivity** – Does an entity typically have just one such value or is the value non-exclusive? For example, people have just one SSN and DOB, but they may have multiple phone numbers.
- **Stability** – Is this an exclusive value that is generally constant over an entity's lifetime, or can it change over time? For example, a person's SSN and DOB remain consistent whereas their street address can change.

The software comes preconfigured with the common attributes and expected behaviors of people and organizations. The user can start loading and resolving entities without any configuration, training, or tuning. If there is a need to add a new attribute, such as an additional identifier, the user can just add the name of the attribute and assign its three behaviors.

For additional insights to Senzing Principle Based Entity Resolution, please see https://senzing.com/wp-content/uploads/Principle-Based-Entity-Resolution-092519.pdf.

Veeva SafeMine Tokenization

Like other PPRLS solutions, Veeva SafeMine supports de-identification of personal data by generating anonymized tokens (i.e., cryptographic hashes) of different permutations of PII which have been deemed to be uniquely identifying. These tokens are generated at each data source and then combined at a central location all in accordance with a HIPAA expert determination.

Different from other PPRLS, SafeMine takes a unique approach by aggregating each patient's data, which may be spread across several records, before it de-identifies the patient as a whole. This process uses fully identifiable PII and captures the full patient record over time, unlike the one record at a time approach used by other PPRLS. The data are aggregated securely behind the data supplier's firewall in a covered entity environment, and it allows SafeMine to create multiple versions of each token for a patient, based on the variations in the PII that were observed across all the records for a patient, in a way that is not possible with more conventional record-based tokens. Only once this aggregation process has occurred is patient data de-identified and output from SafeMine.

The patient data aggregation process is very powerful. In the real world, almost all health datasets will have multiple records for each patient (oftentimes without a reliable internal patient ID). We also know that these records frequently will have significant differences in PII whether due to data entry errors (i.e., misspellings, entering the wrong value in a field, etc.) or due to life events (i.e., last name changing due to marriage, change in address/city/state/zip due to moving, etc.). Matching on de-identified tokens will oftentimes either miss a match due to not being able to account for discrepancies in PII (false negative) or conversely it can make an erroneous match due to matching without considering the full set of PII available (false positive). The SafeMine data aggregation process significantly reduces the occurrences of these false positive and false negatives.

The patient data aggregation process is configurable and can be tailored to the individual dataset depending on which fields are available (ex: if the dataset contains a strong ID or not) and whether the goal is to minimize false positives vs false negatives.

The SafeMine software automatically generates all the tokens that are permissible to be created per Veeva's HIPAA expert determination (assuming the relevant fields are available and populated with data). The software handles HIPAA compliance through technology. All HIPAA restrictions are baked into the software itself ensuring privacy sensitive data never leaves partner sites. The software itself and all outputs are also fully auditable by data suppliers.

The Veeva SafeMine de-identification engine is Java-based and operates from the command line. Running on Java allows the software to be deployed in a wide variety of environments including cloud-based systems such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud. On-prem Windows and Linux distributions are also available. SafeMine v1.1804 was the software version used in this evaluation.

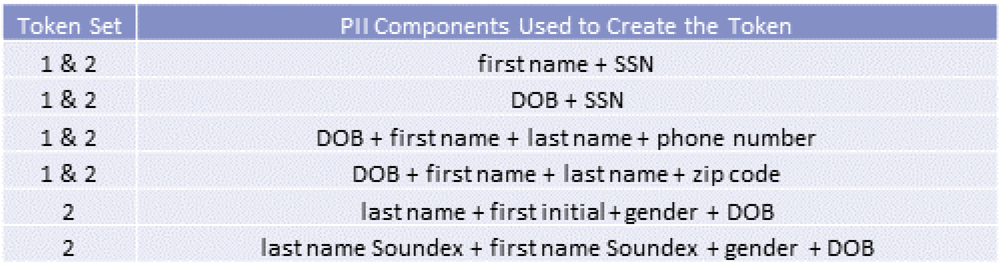
The de-identification engine outputs the patient tokens in a comma-delimited file format. Once the files have been created the tokens are matched using the SafeMine matching software. We did not license the matching software for this evaluation and opted to send Veeva the files instead, where they were matched behind Veeva's firewall.

Veeva SafeMine Matching

Once the tokens have been extracted from each data supplier, they are compared to each other for matching. Any match between two tokens is considered a patient match. As explained above, a patient may have multiple versions of the same token owing to the SafeMine patient data aggregation process. This allows token matching to account for situations where there are discrepancies in PII between two different datasets. For example, a patient may have both her maiden name and married name recorded in Database A and only her maiden name recorded in Database B. Because SafeMine aggregated the patient's data in Database A, this patient will have two versions of every token that uses Last Name (one with her maiden name and one with her married name). This enables matching of the full Database A patient record to the record in Database B (including any records in Database A where the married name was used). The same idea holds true for differences due to typos, patient moving, etc.

The set of tokens that are used to match patients between the datasets is fully customizable. For example, if a user wants to limit the number of false positive matches, Veeva will match the data using a token set that is comprised of tokens that require exact matches on multiple fields. Likewise, if a user wants to minimize the number of false negatives, Veeva will match the data using a token set that contains tokens which allow for some fuzzy matching to take place. We evaluated the performance of two token sets for this analysis: one which sought to minimize false positives and another which sought to minimize false negatives. For a more detailed description of the tokens and token sets that were used in this evaluation, see Methods: Table 6.

Methods Table 6 – Veeva SafeMine Tokens Created



Token Set	PII Components Used to Create the Token
1 & 2	first name + SSN
1 & 2	DOB + SSN
1 & 2	DOB + first name + last name + phone number
1 & 2	DOB + first name + last name + zip code
2	last name + first initial + gender + DOB
2	last name Soundex + first name Soundex + gender + DOB

The token matching process is recursive in that it will continue linking patients until no additional linkage is possible. For example, if Patient A matches to Patient B on one token and Patient B matches to Patient C on another token, the system will recursively keep matching until Patient C also matches to Patient A (even if they do not share any identical tokens).

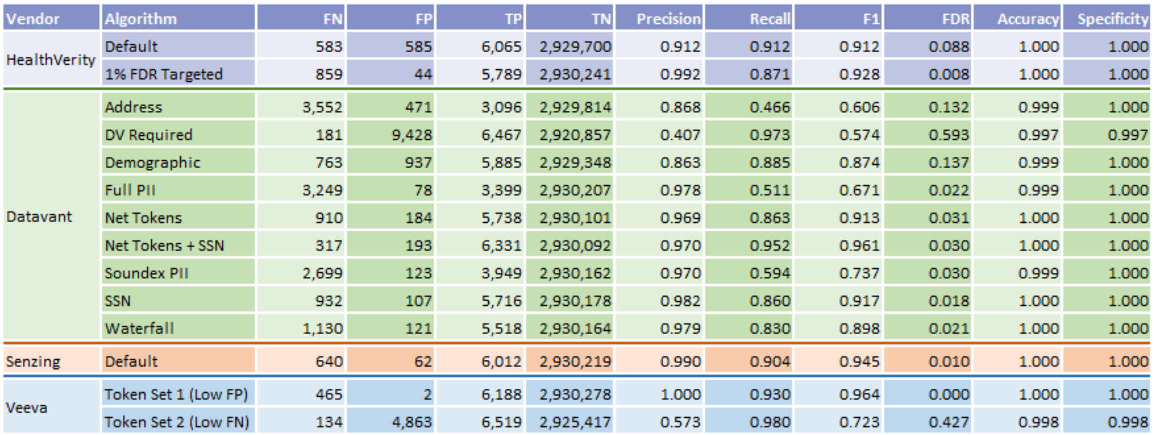
The matching system is also not limited to matching just two datasets; it can be used to match an unlimited number of datasets simultaneously (crucial because typically several datasets are being matched at once).

RESULTS

Scenario 1 – Linkage with High Data Quality and Completeness

F1 scores for the linkage between the six SEER registries ranged from 0.574 to 0.964. The number of true positive pairs ranged from 3,096 to 6,519 and the number of false positive pairs ranged from 2 to 9,428. These and other quantitative metrics are presented in *Results: Table 1A*.

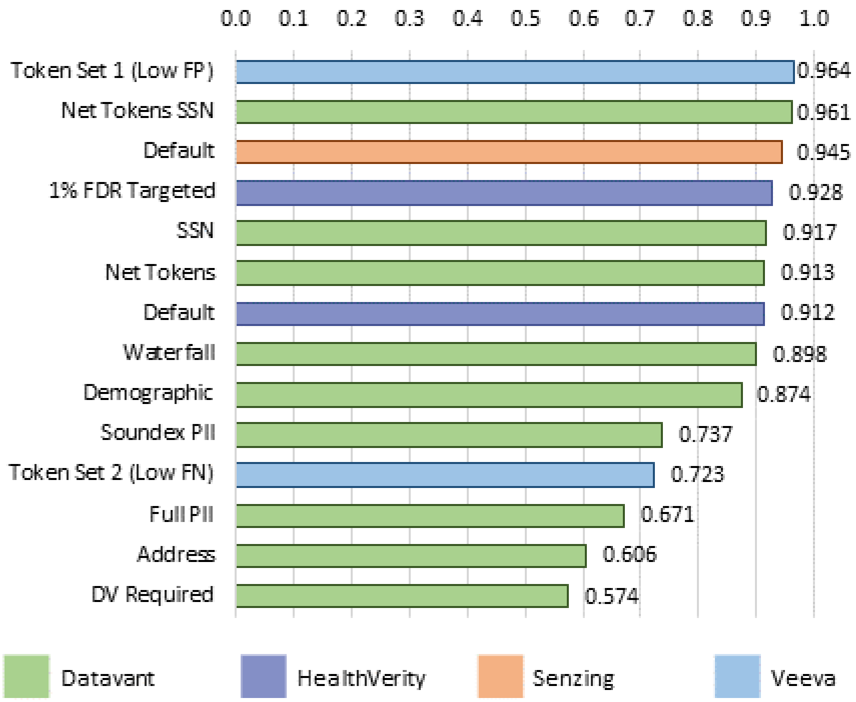
Results: Table 1A – Quantitative Metrics for Linkage Scenario 1 (6 SEER Registries, High Data Quality)



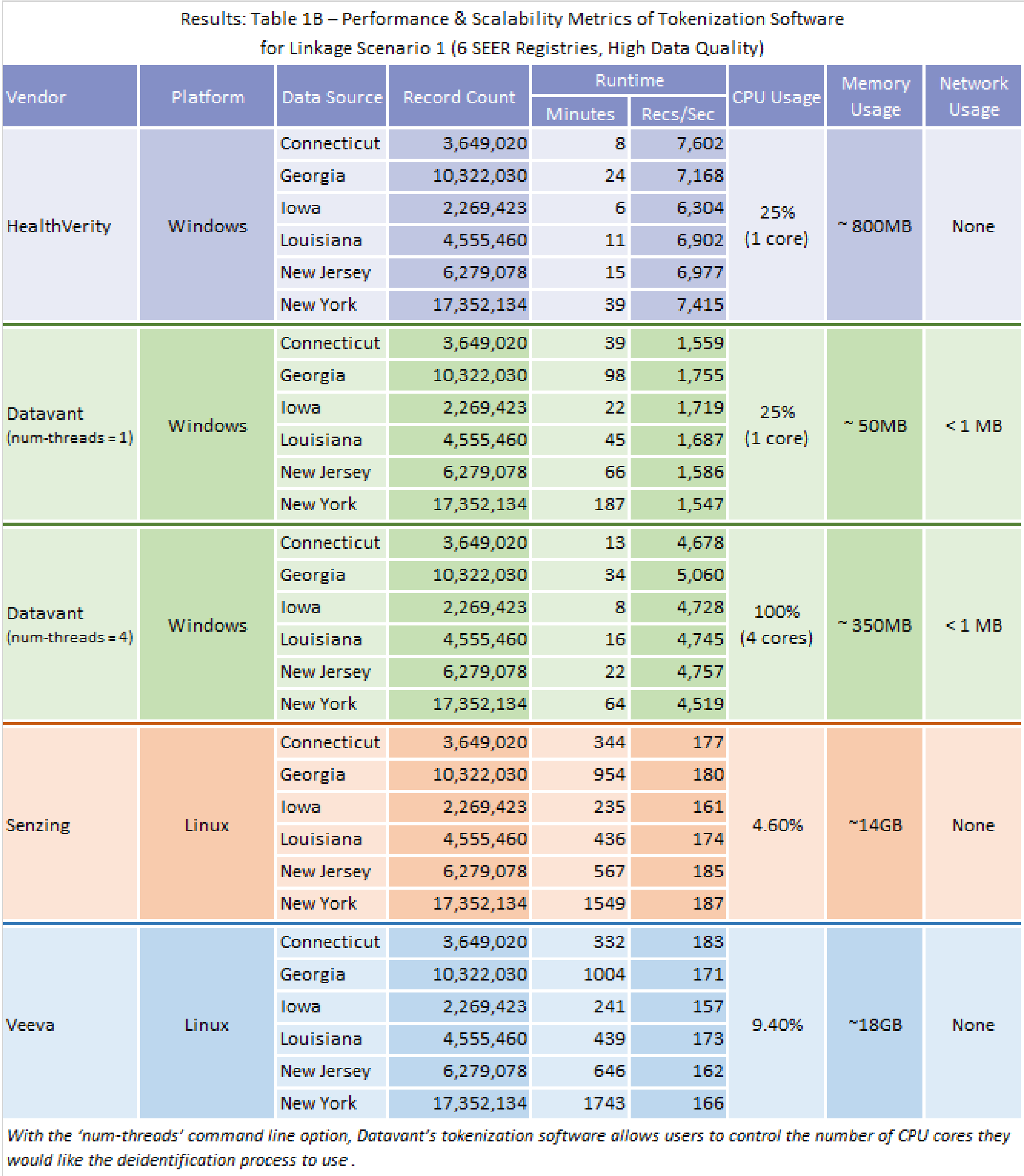
Vendor	Algorithm	FN	FP	TP	TN	Precision	Recall	F1	FDR	Accuracy	Specificity
HealthVerity	Default	583	585	6,065	2,929,700	0.912	0.912	0.912	0.088	1.000	1.000
	1% FDR Targeted	859	44	5,789	2,930,241	0.992	0.871	0.928	0.008	1.000	1.000
Datavant	Address	3,552	471	3,096	2,929,814	0.868	0.466	0.606	0.132	0.999	1.000
	DV Required	181	9,428	6,467	2,920,857	0.407	0.973	0.574	0.593	0.997	0.997
	Demographic	763	937	5,885	2,929,348	0.863	0.885	0.874	0.137	0.999	1.000
	Full PII	3,249	78	3,399	2,930,207	0.978	0.511	0.671	0.022	0.999	1.000
	Net Tokens	910	184	5,738	2,930,101	0.969	0.863	0.913	0.031	1.000	1.000
	Net Tokens + SSN	317	193	6,331	2,930,092	0.970	0.952	0.961	0.030	1.000	1.000
	Soundex PII	2,699	123	3,949	2,930,162	0.970	0.594	0.737	0.030	0.999	1.000
	SSN	932	107	5,716	2,930,178	0.982	0.860	0.917	0.018	1.000	1.000
	Waterfall	1,130	121	5,518	2,930,164	0.979	0.830	0.898	0.021	1.000	1.000
Senzing	Default	640	62	6,012	2,930,219	0.990	0.904	0.945	0.010	1.000	1.000
Veeva	Token Set 1 (Low FP)	465	2	6,188	2,930,278	1.000	0.930	0.964	0.000	1.000	1.000
	Token Set 2 (Low FN)	134	4,863	6,519	2,925,417	0.573	0.980	0.723	0.427	0.998	0.998

The Veeva token set that attempted to minimize false positives (token set 1) earned the highest F1 score, followed very closely by Datavant's *Net Tokens + SSN* algorithm. Datavant's *Datavant Required* algorithm earned the lowest F1 score. The F1 scores are ranked from highest to lowest in *Results: Figure 1A*.

Results: Figure 1A – F1 Score Ranking for Linkage Scenario 1
(6 SEER Registries, High Data Quality)



The Windows-based tokenization software processed between 1,547 and 7,602 records per second. CPU usage for the Windows software ranged from 25% (a single CPU core) to 100% (four CPU cores). Memory usage ranged from 50-800 MB. The Linux-based tokenization software processed between 157 and 187 records per second. CPU usage for the Linux software ranged from 4.6-9.4%. Memory usage ranged from 14-18GB. Performance and scalability metrics for the tokenization utilities are presented in *Results: Table 1B*.



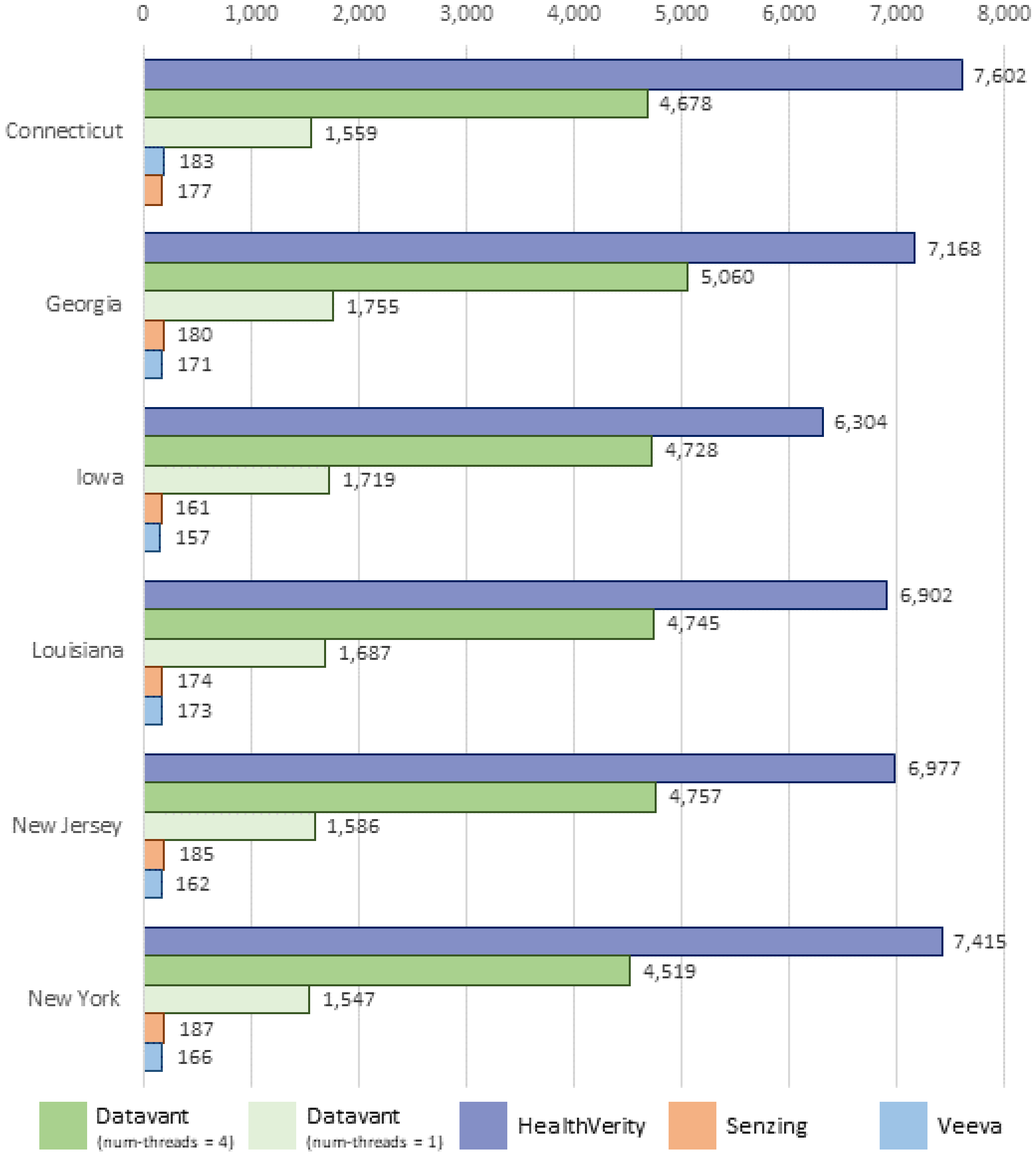
Results: Table 1B – Performance & Scalability Metrics of Tokenization Software
for Linkage Scenario 1 (6 SEER Registries, High Data Quality)

Vendor	Platform	Data Source	Record Count	Runtime		CPU Usage	Memory Usage	Network Usage
				Minutes	Recs/Sec			
HealthVerity	Windows	Connecticut	3,649,020	8	7,602	25% (1 core)	~ 800MB	None
		Georgia	10,322,030	24	7,168			
		Iowa	2,269,423	6	6,304			
		Louisiana	4,555,460	11	6,902			
		New Jersey	6,279,078	15	6,977			
		New York	17,352,134	39	7,415			
Datavant (num-threads = 1)	Windows	Connecticut	3,649,020	39	1,559	25% (1 core)	~ 50MB	< 1 MB
		Georgia	10,322,030	98	1,755			
		Iowa	2,269,423	22	1,719			
		Louisiana	4,555,460	45	1,687			
		New Jersey	6,279,078	66	1,586			
		New York	17,352,134	187	1,547			
Datavant (num-threads = 4)	Windows	Connecticut	3,649,020	13	4,678	100% (4 cores)	~ 350MB	< 1 MB
		Georgia	10,322,030	34	5,060			
		Iowa	2,269,423	8	4,728			
		Louisiana	4,555,460	16	4,745			
		New Jersey	6,279,078	22	4,757			
		New York	17,352,134	64	4,519			
Senzing	Linux	Connecticut	3,649,020	344	177	4.60%	~14GB	None
		Georgia	10,322,030	954	180			
		Iowa	2,269,423	235	161			
		Louisiana	4,555,460	436	174			
		New Jersey	6,279,078	567	185			
		New York	17,352,134	1549	187			
Veeva	Linux	Connecticut	3,649,020	332	183	9.40%	~18GB	None
		Georgia	10,322,030	1004	171			
		Iowa	2,269,423	241	157			
		Louisiana	4,555,460	439	173			
		New Jersey	6,279,078	646	162			
		New York	17,352,134	1743	166			

With the 'num-threads' command line option, Datavant's tokenization software allows users to control the number of CPU cores they would like the deidentification process to use .

HealthVerity's tokenization software performed the fastest in our testing. The Linux-based software packages were substantially slower than the Windows-based alternatives even though they were deployed on servers with specifications that exceeded each vendor's recommendations. The processing speeds for the tokenization utilities are visually represented in *Results: Figure 1B*.

Results: Figure 1B – Processing Speed (Records per Second) of the Tokenization Utilities for Linkage Scenario 1 (6 SEER Registries, High Data Quality)



With the 'num-threads' command line option, Datavant's tokenization software allows users to control the number of CPU cores they would like the deidentification process to use.

Both HealthVerity and Veeva conducted the matching behind their firewalls, which was outside of our testing environment, so we were only able to collect performance and scalability metrics for the Datavant and Senzing matching software. The Datavant software, which is Windows-based, processed between 1,423 and 2,857 records per second depending on the matching algorithm that was used. CPU usage for the Datavant software never exceeded 25% (a single CPU core). Memory usage for the software consistently averaged around 54 GB. The Senzing software, which is Linux-based, processed 320 records per second. The CPU usage for Senzing's software was 99%, taking full advantage of the CPU resources that were available. The Senzing software used approximately 107GB of memory to perform the matching. Performance and scalability metrics for these utilities are presented in *Results: Table 1C*.

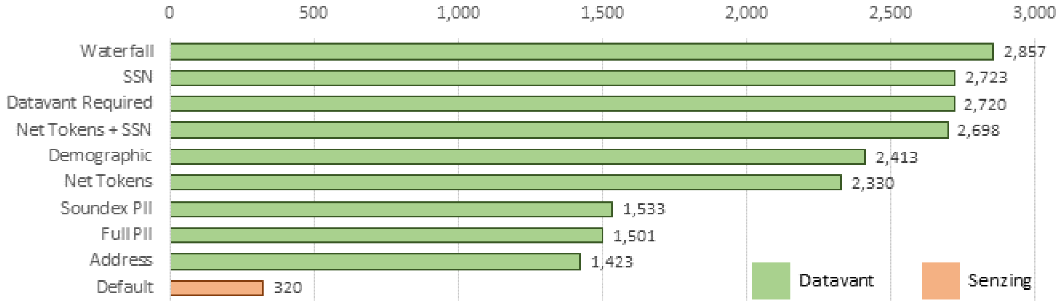
Results: Table 1C – Performance & Scalability Metrics of Matching Software
for Linkage Scenario 1(6 SEER Registries, High Data Quality)

Vendor	Platform	Record Count	Algorithm	Runtime		CPU	Memory	Network
				Minutes	Recs/Sec	Usage	Usage	Usage
HealthVerity	Windows	44,427,148	Default	Unknown	Unknown	Unknown	Unknown	Unknown
			1% FDR Targeted					
Datavant	Windows	44,427,148	Address	520	1,423	25% (1 core)	~54GB	None
			Datavant Required	272	2,720			
			Demographic	307	2,413			
			Full PII	493	1,501			
			Net Tokens	318	2,330			
			Net Tokens SSN	274	2,698			
			Soundex PII	483	1,533			
			SSN	272	2,723			
			Waterfall	259	2,857			
Senzing	Linux	44,427,148	Default	2,316	320	99%	~107GB	None
Veeva	Linux	44,427,148	Token Set 1 (Low FP)	Unknown	Unknown	Unknown	Unknown	Unknown
			Token Set 2 (Low FN)					

Performance and scalability metrics were not collected for HealthVerity or Veeva because the matching took place off-site, behind each vendor's respective firewalls.

Datavant's Waterfall algorithm linked records the fastest in our testing. The Senzing software performed the slowest. Once again, there was a large gap in terms of speed between the Windows and Linux software packages. The processing speeds for the matching utilities are visually represented in *Results: Figure 1C*.

Results: Figure 1C – Processing Speed (Records per Second) of the Matching Utilities
for Linkage Scenario 1 (6 SEER Registries, High Data Quality)



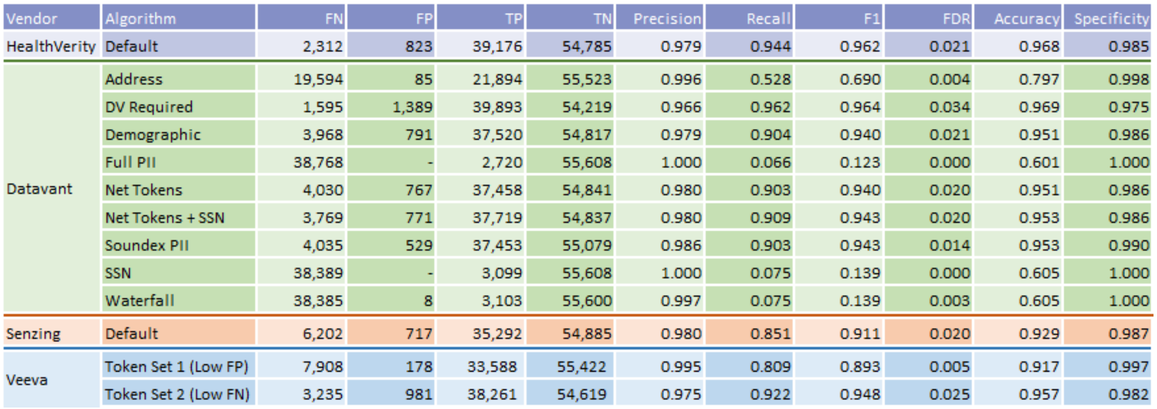
Performance and scalability metrics were not collected for HealthVerity and Veeva because the matching took place off-site, behind each vendor's firewall.

It wasn't possible to link all 6 registry files at the same time using Datavant's matching software due to memory constraints. The 6 files were broken down into 3 sets of files and the aggregate match result was obtained by combining the match results from the 3 sets and then deduplicating the links. The numbers shown above represent the average processing speed for each matching algorithm across the 3 sets of files.

Scenario 2 – Linkage with Moderate Data Quality and Completeness

F1 scores for the linkage between the GA registry and CVS/Walgreens ranged from 0.123 to 0.964. The number of true positive pairs ranged from 2,720 to 39,893, and the number of false positive pairs ranged from 0 to 1,389. These and other quantitative metrics are presented in *Results: Table 2A*.

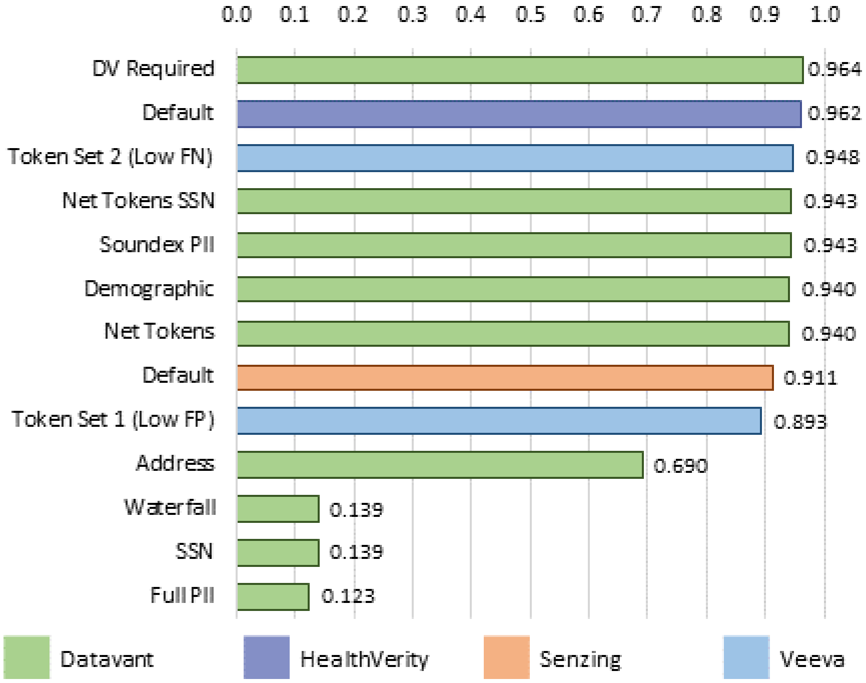
Results: Table 2A – Quantitative Metrics for Linkage Scenario 2 (GA Registry & CVS/Walgreens, Moderate Data Quality)



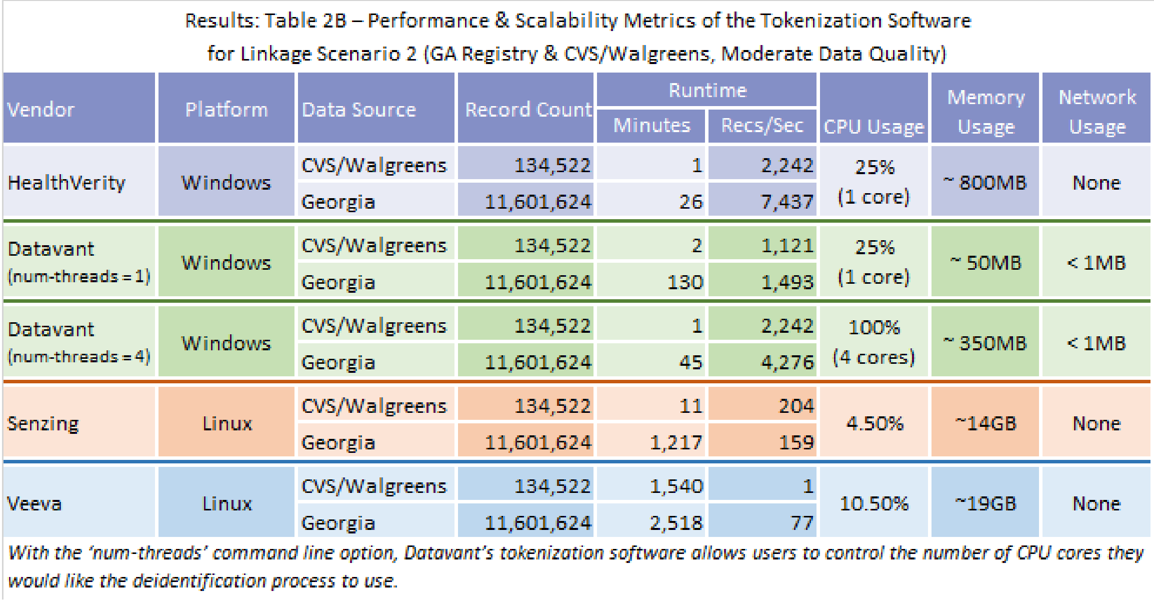
Vendor	Algorithm	FN	FP	TP	TN	Precision	Recall	F1	FDR	Accuracy	Specificity
HealthVerity	Default	2,312	823	39,176	54,785	0.979	0.944	0.962	0.021	0.968	0.985
Datavant	Address	19,594	85	21,894	55,523	0.996	0.528	0.690	0.004	0.797	0.998
	DV Required	1,595	1,389	39,893	54,219	0.966	0.962	0.964	0.034	0.969	0.975
	Demographic	3,968	791	37,520	54,817	0.979	0.904	0.940	0.021	0.951	0.986
	Full PII	38,768	-	2,720	55,608	1.000	0.066	0.123	0.000	0.601	1.000
	Net Tokens	4,030	767	37,458	54,841	0.980	0.903	0.940	0.020	0.951	0.986
	Net Tokens + SSN	3,769	771	37,719	54,837	0.980	0.909	0.943	0.020	0.953	0.986
	Soundex PII	4,035	529	37,453	55,079	0.986	0.903	0.943	0.014	0.953	0.990
	SSN	38,389	-	3,099	55,608	1.000	0.075	0.139	0.000	0.605	1.000
	Waterfall	38,385	8	3,103	55,600	0.997	0.075	0.139	0.003	0.605	1.000
Senzing	Default	6,202	717	35,292	54,885	0.980	0.851	0.911	0.020	0.929	0.987
Veeva	Token Set 1 (Low FP)	7,908	178	33,588	55,422	0.995	0.809	0.893	0.005	0.917	0.997
	Token Set 2 (Low FN)	3,235	981	38,261	54,619	0.975	0.922	0.948	0.025	0.957	0.982

Datavant's *Datavant Required* algorithm earned the highest F1 score, followed very closely by HealthVerity. Datavant's *Full PII* algorithm earned the lowest F1 score. The F1 scores were ranked from highest to lowest in *Results: Figure 2A*.

Results: Figure 2A – F1 Score Ranking for Linkage Scenario 2 (GA Registry & CVS/Walgreens, Moderate Data Quality)



The Windows-based tokenization software processed between 1,121 and 7,437 records per second. CPU usage for the Windows software ranged from 25% (a single CPU core) to 100% (four CPU cores). Memory usage ranged from 50-800 MB. The Linux-based tokenization software processed between 1 to 204 records per second. CPU usage for the Linux software ranged from 4.5% to 10.5%. Memory usage ranged from 14-19 GB. Performance and scalability metrics for these utilities are presented in *Results: Table 2B*.



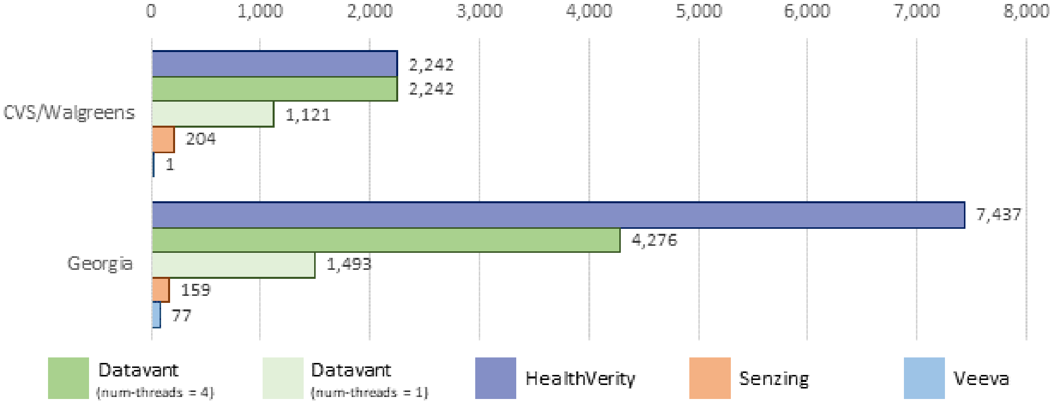
Results: Table 2B – Performance & Scalability Metrics of the Tokenization Software for Linkage Scenario 2 (GA Registry & CVS/Walgreens, Moderate Data Quality)

Vendor	Platform	Data Source	Record Count	Runtime		CPU Usage	Memory Usage	Network Usage
				Minutes	Recs/Sec			
HealthVerity	Windows	CVS/Walgreens	134,522	1	2,242	25% (1 core)	~ 800MB	None
		Georgia	11,601,624	26	7,437			
Datavant (num-threads = 1)	Windows	CVS/Walgreens	134,522	2	1,121	25% (1 core)	~ 50MB	< 1MB
		Georgia	11,601,624	130	1,493			
Datavant (num-threads = 4)	Windows	CVS/Walgreens	134,522	1	2,242	100% (4 cores)	~ 350MB	< 1MB
		Georgia	11,601,624	45	4,276			
Senzing	Linux	CVS/Walgreens	134,522	11	204	4.50%	~14GB	None
		Georgia	11,601,624	1,217	159			
Veeva	Linux	CVS/Walgreens	134,522	1,540	1	10.50%	~19GB	None
		Georgia	11,601,624	2,518	77			

With the 'num-threads' command line option, Datavant's tokenization software allows users to control the number of CPU cores they would like the deidentification process to use.

HealthVerity's tokenization software performed the fastest in our testing. The Linux-based software packages were substantially slower than the Windows-based alternatives even though they were deployed on servers with specifications that exceeded each vendor's recommendations. The processing speeds for the tokenization utilities are visually represented in *Results: Figure 2B*.

Results: Figure 2B – Processing Speed (Records per Second) of the Tokenization Utilities for Linkage Scenario 2 (GA Registry & CVS/Walgreens, Moderate Data Quality)



With the 'num-threads' command line option, Datavant's tokenization software allows users to control the number of CPU cores they would like the deidentification process to use.

Both HealthVerity and Veeva conducted the matching behind their firewalls, which was outside of our testing environment, so we were only able to collect performance and scalability metrics for the Datavant and Senzing matching software. The Datavant software, which is Windows-based, processed between 2,406 and 6,722 records per second. CPU usage for the Datavant software never exceeded 25% (a single CPU core). Memory usage was consistent across all our testing, averaging at around 49 GB. The Senzing software, which was Linux-based, processed 325 records per second. The CPU usage for Senzing's software was 99%, indicating the software took full advantage of the CPU resources that were available. The Senzing software used approximately 126 GB to perform the matching. Performance and scalability metrics for these utilities are presented in *Results: Table 2C*.

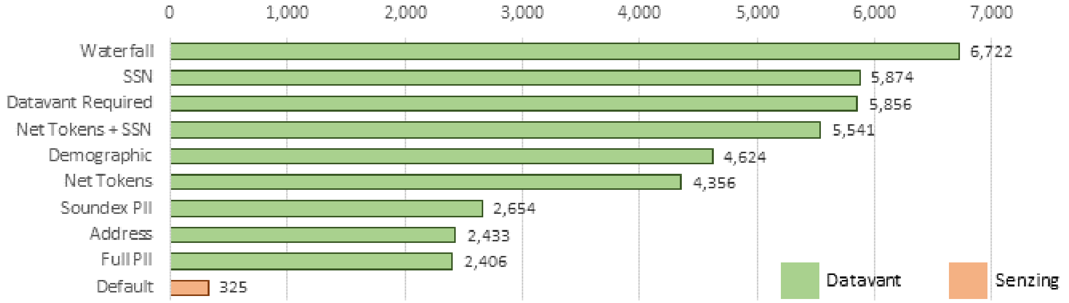
Results: Table 2C – Performance & Scalability Metrics of Matching Software
for Linkage Scenario 2 (GA Registry & CVS/Walgreens, Moderate Data Quality)

Vendor	Platform	Record Count	Algorithm	Runtime		CPU Usage	Memory Usage	Network Usage
				Minutes	Recs/Sec			
HealthVerity	Windows	11,736,146	Default	Unknown	Unknown	Unknown	Unknown	Unknown
Datavant	Windows	11,736,146	Address	80	2,433	25% (1 core)	~49GB	None
			Datavant Required	33	5,856			
			Demographic	42	4,624			
			Full PII	81	2,406			
			Net Tokens	45	4,356			
			Net Tokens SSN	35	5,541			
			Soundex PII	74	2,654			
			SSN	33	5,874			
			Waterfall	29	6,722			
Senzing	Linux	11,736,146	Default	602	325	99%	~126GB	None
Veeva	Linux	11,736,146	Token Set 1 (Low FP) Token Set 2 (Low FN)	Unknown	Unknown	Unknown	Unknown	Unknown

Performance and scalability metrics were not collected for HealthVerity and Veeva because the matching took place off-site, behind each vendor's respective firewalls.

Datavant's *Waterfall* algorithm linked records the fastest in our testing. The Senzing software performed the slowest. Once again, there was a large gap in terms of speed between the Windows and Linux software packages. The processing speeds for the matching utilities are visually represented in *Results: Figure 2C*.

Results: Figure 2C – Processing Speed (Records per Second) of the Matching Utilities
for Linkage Scenario 2 (GA Registry & CVS/Walgreens, Moderate Data Quality)

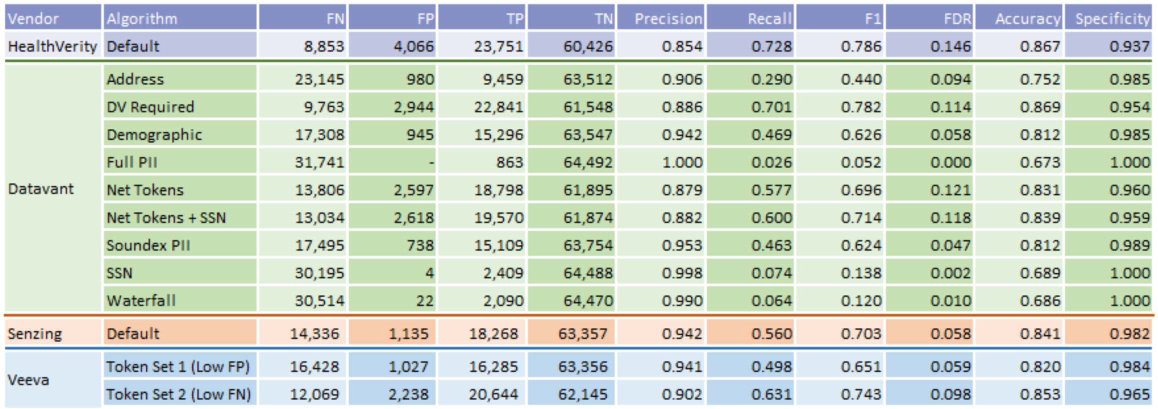


Performance and scalability metrics were not collected for HealthVerity and Veeva because the matching took place off-site, behind each vendor's firewall.

Scenario 3 – Linkage with Low Data Quality and Completeness

F1 scores for the linkage between the modified GA registry data and the modified CVS/Walgreens data ranged from 0.052 to 0.786. The number of true positive pairs ranged from 863 to 23,751, and the number of false positive pairs ranged from 0 to 4,066. These and other quantitative metrics are presented in *Results: Table 3A*.

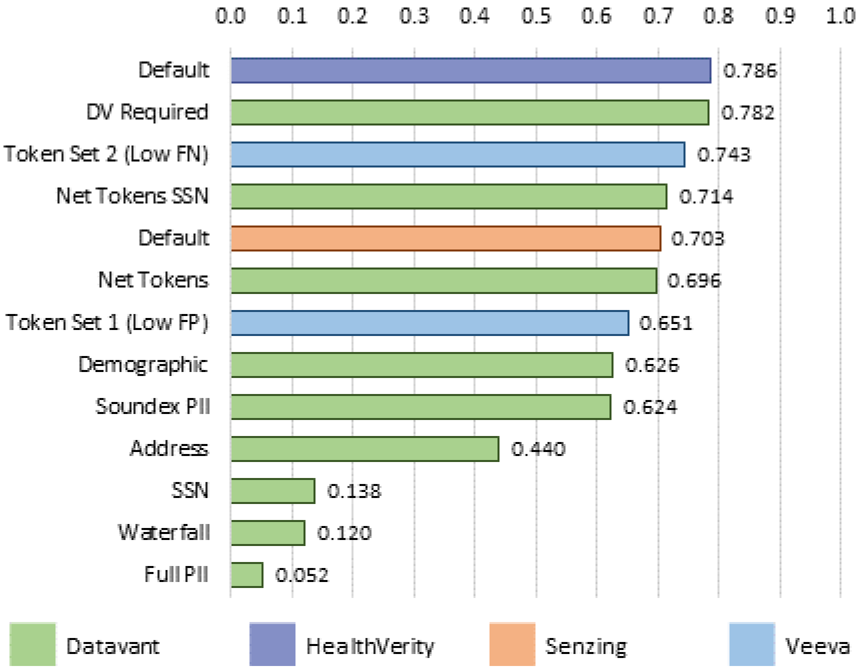
Results: Table 3A – Quantitative Metrics for Linkage Scenario 3 (GA Registry & CVS/Walgreens, Modified PII, Low Data Quality)



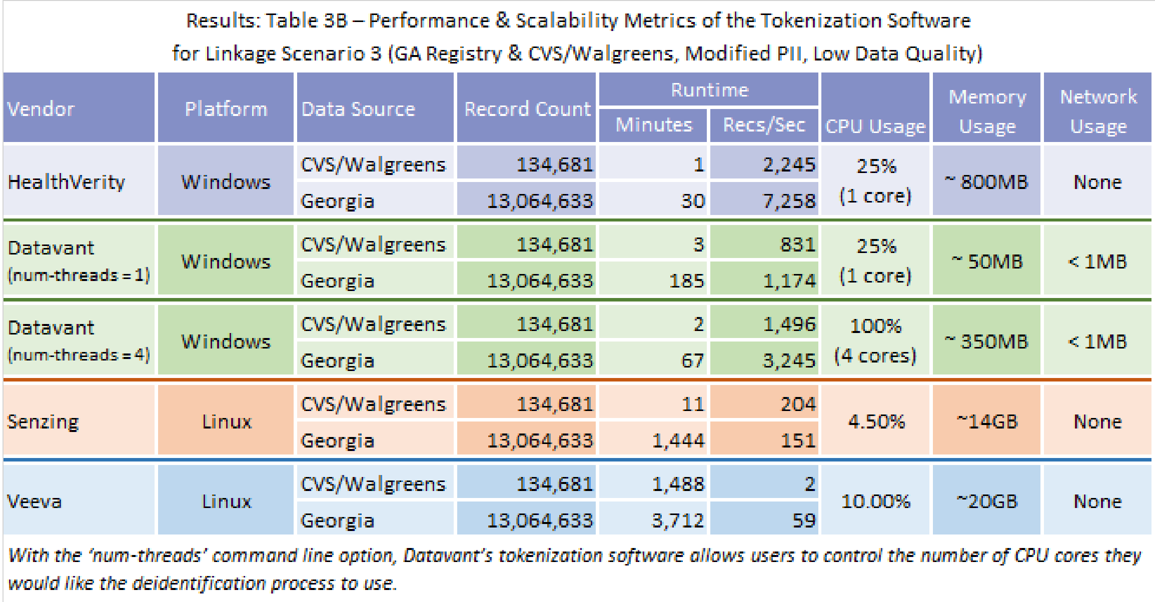
Vendor	Algorithm	FN	FP	TP	TN	Precision	Recall	F1	FDR	Accuracy	Specificity
HealthVerity	Default	8,853	4,066	23,751	60,426	0.854	0.728	0.786	0.146	0.867	0.937
Datavant	Address	23,145	980	9,459	63,512	0.906	0.290	0.440	0.094	0.752	0.985
	DV Required	9,763	2,944	22,841	61,548	0.886	0.701	0.782	0.114	0.869	0.954
	Demographic	17,308	945	15,296	63,547	0.942	0.469	0.626	0.058	0.812	0.985
	Full PII	31,741	-	863	64,492	1.000	0.026	0.052	0.000	0.673	1.000
	Net Tokens	13,806	2,597	18,798	61,895	0.879	0.577	0.696	0.121	0.831	0.960
	Net Tokens + SSN	13,034	2,618	19,570	61,874	0.882	0.600	0.714	0.118	0.839	0.959
	Soundex PII	17,495	738	15,109	63,754	0.953	0.463	0.624	0.047	0.812	0.989
	SSN	30,195	4	2,409	64,488	0.998	0.074	0.138	0.002	0.689	1.000
	Waterfall	30,514	22	2,090	64,470	0.990	0.064	0.120	0.010	0.686	1.000
Senzing	Default	14,336	1,135	18,268	63,357	0.942	0.560	0.703	0.058	0.841	0.982
Veeva	Token Set 1 (Low FP)	16,428	1,027	16,285	63,356	0.941	0.498	0.651	0.059	0.820	0.984
	Token Set 2 (Low FN)	12,069	2,238	20,644	62,145	0.902	0.631	0.743	0.098	0.853	0.965

HealthVerity earned the highest F1 score, followed very closely by Datavant's *Datavant Required* algorithm. Datavant's *Full PII* algorithm earned the lowest F1 score. The F1 scores were ranked from highest to lowest in *Results: Figure 3A*.

Results: Figure 3A – F1 Score Ranking for Linkage Scenario 3
(GA Registry & CVS/Walgreens, Modified PII, Low Data Quality)



The Windows-based tokenization software processed between 831 and 7,258 records per second. CPU usage for the Windows software ranged from 25% (a single CPU core) to 100% (four CPU cores). Memory usage ranged from 50-800 MB. The Linux-based tokenization software processed between 2 and 204 records per second. CPU usage for the Linux software ranged from 4.5-10%. Memory usage ranged from 14-20GB. Performance and scalability metrics for these utilities are presented in *Results: Table 3B*.



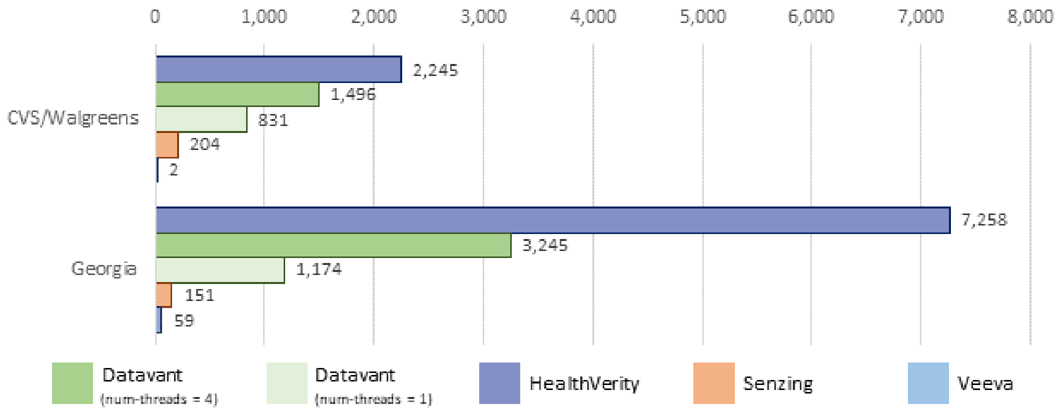
Results: Table 3B – Performance & Scalability Metrics of the Tokenization Software
for Linkage Scenario 3 (GA Registry & CVS/Walgreens, Modified PII, Low Data Quality)

Vendor	Platform	Data Source	Record Count	Runtime		CPU Usage	Memory Usage	Network Usage
				Minutes	Recs/Sec			
HealthVerity	Windows	CVS/Walgreens	134,681	1	2,245	25% (1 core)	~ 800MB	None
		Georgia	13,064,633	30	7,258			
Datavant (num-threads = 1)	Windows	CVS/Walgreens	134,681	3	831	25% (1 core)	~ 50MB	< 1MB
		Georgia	13,064,633	185	1,174			
Datavant (num-threads = 4)	Windows	CVS/Walgreens	134,681	2	1,496	100% (4 cores)	~ 350MB	< 1MB
		Georgia	13,064,633	67	3,245			
Senzing	Linux	CVS/Walgreens	134,681	11	204	4.50%	~14GB	None
		Georgia	13,064,633	1,444	151			
Veeva	Linux	CVS/Walgreens	134,681	1,488	2	10.00%	~20GB	None
		Georgia	13,064,633	3,712	59			

With the 'num-threads' command line option, Datavant's tokenization software allows users to control the number of CPU cores they would like the deidentification process to use.

HealthVerity's tokenization software performed the fastest in our testing. The Linux-based software packages were substantially slower than the Windows-based alternatives even though they were deployed on servers with specifications that exceeded each vendor's recommendations. The processing speeds of the tokenization utilities are visually represented in *Results: Figure 3B*.

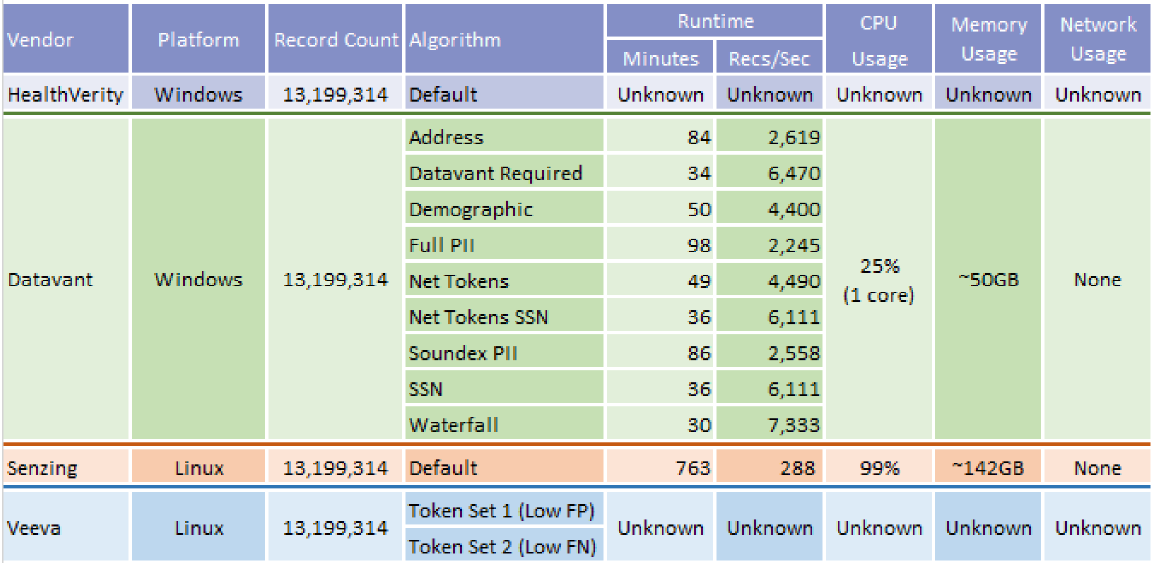
Results: Figure 3B – Processing Speed (Records per Second) of the Tokenization Utilities
for Linkage Scenario 3 (GA Registry & CVS/Walgreens, Modified PII, Low Data Quality)



With the 'num-threads' command line option, Datavant's tokenization software allows users to control the number of CPU cores they would like the deidentification process to use.

Both HealthVerity and Veeva conducted the matching behind their firewalls, which was outside of our testing environment, so we were only able to collect performance and scalability metrics for the Datavant and Senzing matching software. The Datavant software, which runs on Windows, processed between 2,249 and 7,236 records per second. CPU usage for the Datavant software never exceeded 25% (a single CPU core). Memory usage was consistent across all our testing at around 50 GB. The Senzing software, which runs on Linux, processed 288 records per second. The CPU usage for Senzing's software was 99%, which means it took full advantage of the CPU resources that were available. The Senzing software used approximately 142 GB of memory to perform the matching. Performance and scalability metrics for these utilities are presented in *Results: Table 3C*.

Results: Table 3C – Performance & Scalability Metrics of Matching Software
for Linkage Scenario 3 (GA Registry & CVS/Walgreens, Modified PII, Low Data Quality)

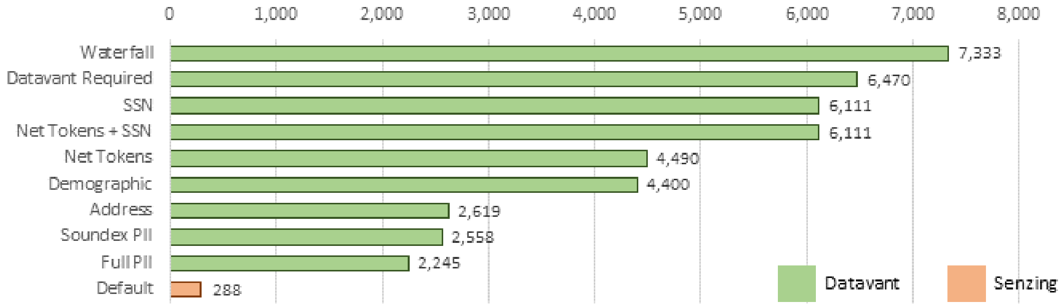


Vendor	Platform	Record Count	Algorithm	Runtime		CPU Usage	Memory Usage	Network Usage
				Minutes	Recs/Sec			
HealthVerity	Windows	13,199,314	Default	Unknown	Unknown	Unknown	Unknown	Unknown
Datavant	Windows	13,199,314	Address	84	2,619	25% (1 core)	~50GB	None
			Datavant Required	34	6,470			
			Demographic	50	4,400			
			Full PII	98	2,245			
			Net Tokens	49	4,490			
			Net Tokens SSN	36	6,111			
			Soundex PII	86	2,558			
			SSN	36	6,111			
			Waterfall	30	7,333			
Senzing	Linux	13,199,314	Default	763	288	99%	~142GB	None
Veeva	Linux	13,199,314	Token Set 1 (Low FP)	Unknown	Unknown	Unknown	Unknown	Unknown
			Token Set 2 (Low FN)					

Performance and scalability metrics were not collected for HealthVerity and Veeva because the matching took place off-site, behind each vendor's respecitive firewalls.

Datavant's Waterfall algorithm linked records the fastest in our testing. The Senzing software performed the slowest. Once again, there was a large gap in terms of speed between the Windows and Linux software packages. The processing speeds of the matching utilities are visually represented in *Results: Figure 3C*.

Results: Figure 3C – Processing Speed (Records per Second) of the Matching Utilities
for Linkage Scenario 3 (GA Registry & CVS/Walgreens, Modified PII, Low Data Quality)



Performance and scalability metrics were not collected for HealthVerity and Veeva because the matching took place off-site, behind each vendor's firewall.

Deployment – HealthVerity and Datavant

The tokenization software developed by HealthVerity was distributed as a stand-alone Java jar file. The tokenization and matching software developed by Datavant were distributed as a stand-alone executable. Both applications were very easy to deploy and were up and running in a very short amount of time, due in large part to their stand-alone nature and the Windows environment.

Deployment – Senzing

The process of deploying the Senzing software was very straightforward and well documented. It's distributed as a Linux package and the process of installing a package should be second nature to anyone familiar with Linux.

Senzing provides instructions for the installation of the software and Postgres configuration directly on their website, including tuning the database for high input/output performance. The installation and configuration instructions were straightforward and did not present any obstacles.

The Senzing software was deployed on an HP ProLiant DL360 Gen9 server chassis with the following components:

- Two Intel Xeon E5-2650 v3 CPUs operating at 2.30GHz with hyperthreading enabled so that they are presented to the Linux operating system as 40 CPUs.
- 256 GBs of memory.
- A direct-attached Samsung consumer grade 1TB SSD.
- The Linux kernel of 3.10.0-1160.53.1 as part of the CentOS 7 distribution.
- Postgres version 12.9
- Additional software requirements as noted by Senzing, including softhsm2 which is used for hashing data, and various other supporting python modules.

The server was sized based on recommendations provided by Senzing. Senzing sizes servers based on the expected number of records to be processed. This includes their recommendations for CPU cores, memory, and storage; as well as hosting the Postgres database on a solid-state hard drive physically attached to the same server as the Senzing software. Senzing did not provide specific CPU requirements beyond the number of cores, and they did not specify requirements for the SSD aside from size. The server as built exceeds the CPU core and memory requirements specified by Senzing.

The hashing of data required the user to maintain an active connection for a long period of time and to respond to a command prompt during the process after long wait periods. This conflicted with IMS' idle time disconnect policy. To resolve the issue, IMS were forced to disable the idle time disconnect for the server.

The Senzing software was bundled with an outdated version of Java. After consulting with Senzing, this outdated version of Java was replaced with the latest release that was available at the time of the evaluation.

The requirements of a physically attached solid state hard drive forced IMS to deviate from its preferred practices for server construction. There was not a clear reason given for this requirement versus using network attached/managed storage. During the evaluation period, we did not notice disk access as being a bottleneck and are dubious about the requirement.

Deployment – Veeva SafeMine

The process of getting the SafeMine software up and running was time consuming and raised many issues and questions.

The software was provided to us via a secure file transfer mechanism as a set of compressed files. The prerequisite software came from our own software repositories which are clones from the public repositories of Red Hat. The deployment of the SafeMine software required a script to be run. Upon running the script, one of the first things it did was to remove the controlled repositories that we use. Instead, the software used a set of local files that were delivered as part of the software package. These were older versions of the packages that we already had installed. The newer versions, which were removed, were updated by RedHat for security and bug fixes. The script also disabled all the firewalls and secure Linux components that were in place, which is a violation of IMS' deployment requirements. IMS needed to go back after the fact to update the software libraries for security patches and reenable all of their security protocols.

The initial deployment instructions were to setup a virtual machine (VM) with the SafeMine software and configurations in place, and to then clone that VM nine times so that we would have one VM for each of the ten files being hashed. In normal operations deploying software should generally be straightforward, but in this case, there were many post-deployment steps that we had to go through for the SafeMine software to be able to process the specific data set we were hashing on each server.

The database server ran on an HP Proliant DL325 Gen10 server chassis with the following components:

- A 16-core AMD Epyc 7351p CPU operating at 2.4GHz (2.9GHz boost) with hyperthreading enabled so the 16 cores can do the work of 32 cores.
- 256 GBs of memory.
- The Linux kernel of 3.10.0-1160.53.1 as part of the CentOS 7 distribution.
- MariaDB version 5.7.

Each of the 10 VMs, which were configured to process the data files, had the following components:

- A 10-core Xeon 4114 Silver CPU operating at 2.2GHz (3.0GHz boost) with hyperthreading enabled so the 10 cores can do the work of 20 cores.
- 64 GBs of memory.
- The Linux kernel of 3.10.0-1160.53.1 as part of the CentOS 7 distribution.

The servers were sized based on the recommendations from Veeva.

With regards to the amount and types of resources required for the deployment, the database server requirements seem in line with other products we've used previously, but we feel that it is unreasonable to deploy a server for each file being hashed. By using just one of the application servers we were able to use most of the database's resources and with two of the application servers running at the same time we completely saturated the database.

Ease of Use

Each of the applications tested in this evaluation were run from the command line. Command-line applications require more memorization and familiarity to operate than applications with GUIs, which are typically more intuitive and easier to use.

HealthVerity, Datavant, and Senzing provided user manuals that summarized the command-line options for their respective tools and included examples of how those options should be used. These manuals were very helpful during the evaluation process, as even after using each of the applications for several weeks we would still need to refer to them from time to time.

Due to how the deployment was setup, and how the configuration files were put in place, one of IMS' Systems Administrators needed to work directly with Veeva's engineering team to do the work for this project, as the files were stored in locations that required root access to the server.

All-in-all, it required several phone calls with Veeva's team to tune each server's configuration file to get the output to work properly. We generally feel like deploying and using the Veeva SafeMine software would be impossible without working directly with Veeva's engineering team, as very little documentation was provided beyond the resources required.

Though not strictly necessary, it would be advantageous nonetheless if each of the vendors we evaluated were to develop GUI-based versions of their tokenization and matching utilities. This would allow non-technical personnel to use the software more easily. Datavant does provide a GUI-based version of its tokenization utility, as mentioned earlier, but this version of the application is less efficient when processing files with many gigabytes of records. The reason behind this limitation is unclear given that the GUI-based application could be designed to call the command-line version using the arguments that the user entered on the GUI. It would be preferable if Datavant would address this issue and develop a GUI-based version of its matching utility to go alongside the tokenization utility.

Pre-Processing

The tokenization utilities from the first two vendors we evaluated (HealthVerity and Datavant) were only capable of accepting one value per identifier type per record. It is not uncommon for patients to have more than one last name, telephone number, or address on file, and quite often, data systems will export all the information onto a single line.

To get around this constraint, we were able to develop a SAS program to restructure the source data that were used in each of the linkages. The SAS program generated all the permutations of last name, telephone number, and address for each patient, and then output each permutation to a separate line alongside the other identifiers such as first name, DOB, and SSN. While this workaround overcame the limitations of these two systems, it may not be feasible for all users to have this capability and represents a limitation in the out-of-the-box use of these applications.

The source data for the SEER registry and pharmacy linkages did not share a common record layout, so we needed to develop two separate SAS programs to restructure each of those datasets. We feel that it is unlikely that many users of the PPRLS software will have the technical skills and/or staff on-hand to write a program to restructure the source data each time they encounter a new record layout to satisfy the constraints of the tokenization utilities.

In addition, the restructuring process significantly increased the number of records, often by a factor of 15 or more, as shown previously in *Methods: Tables 1-3*. Because the same first name, middle name, DOB, SSN, and sex appeared on each record for an individual, this repetition profoundly increased the size of each of the input files. We suspect that it may have also negatively impacted the performance and scalability metrics of the tokenization and matching utilities, as it is quite likely that the utilities performed the same work multiple times across multiple records for the same patient.

The tokenization utilities developed by Veeva and Senzing were capable of handling multiple values per identifier on a single line or multiple values on multiple lines, thus eliminating the need to restructure any records. That said, we opted to use the same set of files (with one value per identifier type per record) in order to try to maintain a level playing field throughout the evaluation process.

Post-Processing - HealthVerity

The HealthVerity PPRLS generates one match result file for each tokenized input file it receives. The match result files are in a delimited file format and each file contains the patient's original ID from the source data alongside a HealthVerity ID. We provided HealthVerity with six tokenized input files for the SEER registry linkage (Scenario 1), two tokenized input files for the pharmacy linkage (Scenario 2), and two tokenized input files for the pharmacy linkage with modified PII (Scenario 3). We received 10 match result files in return.

To identify matches between registries or between the GA registry and CVS/Walgreens, we wrote a SAS program to merge the data in each file using the HealthVerity ID. The thinking behind this is that two patients in separate files who link to the same HealthVerity ID are the same person. We could have also loaded the results into a database and used SQL to perform this work, so that is an option for users who do not have access to SAS. In either case, a person with knowledge of either SAS or databases will be required to perform this step.

The typical use case for HealthVerity involves linking a file to the HealthVerity database and retrieving information from their system, so we understand why they take this approach to generating their results files. It would enhance the end user experience if there were an option to produce a results file that does not rely on use of the HealthVerity ID as an intermediary, but instead produced an output file containing the original IDs from each of the input files that were linked, essentially eliminating the need to use SAS or a database to identify the pairs of linked IDs. This would reduce the expertise necessary to leverage the HealthVerity PPRLS.

Post-Processing – Datavant

The Datavant PPRLS produces a delimited file for its match results. All of the matches are contained within this file. This means we received just one output file for each of the three linkage scenarios, which makes post-processing the match results a little easier.

The first column in the results file contained a group ID, the second column contained the patient's original ID from the source data, and subsequent columns in the file contained the patient's tokens. Individuals who are assigned the same group ID are considered a match, and the records in the file are sorted by group ID and patient ID, which makes it much easier to use the files.

Because the match result files contained all of the tokens for each patient, the size of the files was often an issue. Many of the match result files generated during this evaluation process exceeded eight gigabytes or more. We feel that if users had the option to prevent the tokens from being written to the result files, it would boost the efficiency of the process, ameliorate the file size issue, and improve the overall user experience.

Another issue with the result files, that could be addressed, is the lack of a column containing the filename of the file from which the patient IDs originated. The lack of such a column in the match result files makes it difficult to identify the source of each patient ID in situations where the patient IDs are formatted identically across multiple input files. The inclusion of a column or columns that indicate the ID numbers of the tokens that matched might also prove useful for further study/analysis and refinement of the matching algorithms.

Post-Processing – Senzing

The match results from the Senzing PPRLS are stored in a database until they are needed. They can be viewed and exported using the Senzing software. The software can output several different types of reports, but we were really looking for more of a crosswalk-style report that mapped the linked IDs to one another. We were unable to find this type of report among the default set of reports produced by the software, however, due to the way the Senzing software is set up, it is possible to query the database directly using database administrative software, like PGAdmin. Doing so allows users to create results files that are tailored specifically to their needs. Senzing's support staff assisted us with the creation of a query that was used to generate a set of CSV files that mapped Senzing's internal Resolved Entity ID to the original patient IDs. Individuals who are assigned the same resolved entity ID are considered a match. The format of the file we were able to create was very easy to work with using SAS and was exactly what we needed.

Post-Processing – Veeva SafeMine

After sending the hashed data to Veeva we received an Excel file that listed all the matching ID pairs for each of the linkages in return. This file was very easy to work with and required little-to-no post-processing other than to split the data on each of the tabs into one of three CSV files (one for each linkage). Once the CSV files were created, the process of reading the data using SAS was very trivial.

DISCUSSION

Match Quality – HealthVerity

The HealthVerity PPRLS achieved F1 scores of 0.912 in Scenario 1, 0.962 in Scenario 2, and 0.786 in Scenario 3. As with all PPRLS, the use of the HealthVerity system should be done carefully with a comprehensive understanding of the end users' willingness to accept a certain level of FPs or FNs.

HealthVerity uses one algorithm to match patients to its data system. Their algorithm does not use SSN, as SSN is seldom available to them within the context of their primary objective, which is linking real-world data sources together. We feel that using SSN when it is available would improve their matching algorithm because many of the false negative matches from Scenario 1, and a smaller number of false negative matches from Scenario 2, were among patients who matched on first name, last name, DOB, and SSN.

The false discovery rates for the HealthVerity PPRLS were 8.8% in Scenario 1 initially, 2.1% in Scenario 2, and 14.6% in Scenario 3. These rates, especially those for Scenarios 1 and 2, are limiting for some use cases such as population cancer registry linkages. These scenarios (and Scenario 1 in particular) should be ideal for PPRLS, given the accuracy and completeness of the data, and yet the false discovery rates are 8.8 and 2.1 times higher than desirable.

We followed up with HealthVerity to discuss these findings. They presented us with an option to adjust the sensitivity level of their algorithm to reduce the number of false positives to achieve a false discovery rate of ≤ 1% as a proof of concept. After re-linking the data from Scenario 1 using the new sensitivity level, the F1 score improved from 0.912 to 0.928 and the false discovery rate decreased from 8.8% to 0.8%. Given these results, we believe the HealthVerity PPRLS to be a viable tool for linking individuals in longitudinal studies, provided that the sensitivity level is properly adjusted, and the data are of high quality.

Match Quality – Datavant

The F1 scores for the Datavant PPRLS varied significantly across each of the matching algorithms and linkage scenarios that were tested. A total of nine matching algorithms are available to users of the Datavant PPRLS, and each matching algorithm is designed for a specific use case. The tables of the tokens and their constituent data elements are provided in *Methods: Table 4* on page 11 and *Table 5* on page 12. The default matching algorithm for the Datavant PPRLS is the Net Tokens algorithm.

We feel that the linkage between the six SEER registries (Scenario 1) represents a common use case for the NCI (cohort studies or clinical trials with other data sources such as registries). For these types of uses, the PII for the linkage is likely to feature high levels of accuracy and completeness for each of the identifiers, which may or may not include SSN. In this situation, the Net Tokens + SSN algorithm, which combines the logic of both the Net Tokens and SSN algorithms, earned the highest F1 score. The SSN and Net Tokens algorithms also performed well on their own. However, many data sources will not include SSN, so it is important to recognize the potential value for several of the other tokens such as address and telephone number.

The data that were used to link the GA cancer registry's cases and CVS/Walgreens pharmacy claims in Scenarios 2 and 3 contained very few SSNs. This had a profoundly negative effect on the results obtained for the standalone version of the SSN algorithm. Nevertheless, the Net Tokens and Net Tokens + SSN algorithms remained among the best performing algorithms under these scenarios. The Demographic and Soundex PII algorithms performed well in Scenario 2 but slipped in the rankings in Scenario 3. This may indicate that these algorithms are more sensitive to errors in the data compared to some of the others. The Datavant Required algorithm earned the highest F1 score in Scenarios 2 and 3.

The tokens created for the Datavant Required algorithm represent the token set that Datavant recommends most of its customers generate. Even though the Datavant Required algorithm was the top performer in two of the three scenarios we tested, we cannot recommend it due to its lackluster performance in Scenario 1, where it had an F1 score of just 0.574. The low F1 score was due to the high number of false positives that were identified (9,428 out of 15,895 of the pairs returned). As shown previously in *Methods: Table 4 and Table 5*, the Datavant Required algorithm accepts a match on token 37, which is comprised of a combination of Soundex of first name, Soundex of last name, and DOB. We believed this to be the main point of weakness for this algorithm and a closer examination of the 9,428 false positives bore out our suspicion, with the vast majority having completely different first and last names that coincidentally shared the same Soundex (*e.g.*, the last names Jamison and Jenkins share a common Soundex value of J525). We suspect the use of Soundex was more of an issue in Scenario 1 because of the large pool of records (n=44,427,148) and the fact that the records belonged to patients from across the country. Scenario 2 (n=11,736,145) and Scenario 3 (n=13,199,314) relied on comparatively smaller pools of records that belonged to patients who all reside in Georgia, and the pharmacy records belonged to patients who were prescribed antineoplastic medications, which significantly increased the likelihood of those patients appearing in the GA cancer registry data. The Datavant Required algorithm appears to have an unacceptable risk of FPs for many applications.

Based on the results of our analysis, the Net Tokens + SSN algorithm appears to be the best algorithm compared with others, including the Net Tokens (default) algorithm for population-based linkages. The Net Tokens + SSN algorithm performed well even in situations where the fill rate for SSN was low (Scenario 2), and it outperformed the Net Tokens algorithm in each of the three linkage scenarios we tested.

One potential (but not the only) application for PPRL products is to link data for conducting longitudinal analyses of individuals diagnosed and treated with cancer across the U.S.: in particular, the ability to monitor these patients longitudinally to identify subsequent treatment, recurrence, and treatment of those recurrences. This use case has a very low tolerance for false positive matches, with a false discovery rate of 0-1% being optimal. The false discovery rates for the top performing Datavant algorithms ranged from 1.8%–3.1% in Scenario 1, 1.4%–3.4% in Scenario 2, and 11.4%–12.1% in Scenario 3. The high false discovery rates observed in Scenario 3 are likely due to the modifications made to the identifiers to represent data sources with incomplete, missing, or inaccurate PII values. Scenario 3 and the real-world scenarios it's meant to mimic seem to be less-than-ideal fits for PPRLS, considering the high frequency of typographical errors and missing data present in these (and many real-world) datasets. The false discovery rates for Scenarios 1 and 2 are also less than optimal, given the overall quality and completeness of the original data that were used in those scenarios—and Scenario 1, in particular—as they are between 1.4 and 3.4 times higher than the preferred false discovery rate of ≤ 1%.

We reached out to Datavant to discuss our findings and to determine whether there were options to reduce the number of false positives. A possible solution that was discussed involved the creation of a custom token set, which has been done in other studies that Datavant has supported. This was not possible within the constraints of this evaluation and thus we were unable to explore whether this would improve the results. It is also worth mentioning that the custom tokens aren't necessarily HIPAA-certified and that they would need to be submitted to a HIPAA compliance expert for an audit before they could be used in a linkage in which HIPAA compliance is required.

Similar to HealthVerity, the results of this evaluation reflect a fair comparison of the products that were made available during the evaluation period (July 2020 through June 2021). Datavant has written whitepapers that describe how Datavant leveraged their software in other situations.

Match Quality – Senzing

The Senzing PPRLS achieved F1 scores of 0.945 in Scenario 1, 0.911 in Scenario 2, and 0.703 in Scenario 3. The F1 score that was achieved for Scenario 1 ranked third in our evaluation and was very close to the top two spots (which were nearly tied). The F1 score for Scenario 2 ranked ninth and the F1 score for Scenario 3 ranked fifth. The drop-off in the rankings was mostly caused by an increase in the number of false negative matches.

The false discovery rates for the Senzing PPRLS were 1.0% in Scenario 1, 2.0% in Scenario 2, and 5.8% in Scenario 3. The rates for Scenarios 1 and 3 were better than the averages for those scenarios, which were 11.1% and 7.1% respectively. At 2.0%, the false discovery rate for Scenario 2 was slightly higher than the average (1.4%).

Overall, the results seem to suggest that the Senzing software functions with a high level of specificity, which is ideal in situations where one's tolerance for false positive matches is low.

Match Quality – Veeva SafeMine

We used two sets of tokens to perform the matching when we used the Veeva software: one that aimed to limit the number of false positives (token set 1) and another which sought to limit the number of false negatives (token set 2).

Token set 1 performed very well in Scenario 1, where the data were highly accurate and complete but lagged in Scenarios 2 and 3, where the data were often missing or misspelled. This was in line with our expectations given what we know about these datasets. The F1 scores for token set 1 were 0.964 in Scenario 1, 0.893 in Scenario 2, and 0.651 in Scenario 3. The false discovery rates for token set 1 were 0.03% in Scenario 1, 5% in Scenario 2, 5.9% in Scenario 3. The extremely low false discovery rate in Scenario 1 was very impressive and one of the main reasons why this token set ranked the highest, in terms of its F1 score, across all the software / token sets / algorithms we evaluated.

Token set 2 performed better in Scenarios 2 and 3 than it did in Scenario 1. Again, this was in line with our expectations given what we know about the data. The F1 scores for token set 2 were 0.723 in Scenario 1 (mostly due to the high number of false positives), 0.948 in Scenario 2, and 0.743 in Scenario 3. The false discovery rates for token set 2 were 42.7% in Scenario 1, 2.5% in Scenario 2, and 9.8% in Scenario 3. One of the tokens in this set was based on the Soundex of first and last name. We discussed the issues with creating tokens based on Soundex in the **Match Quality – Datavant** section above. The same issues we discussed there apply here as well.

Performance and Scalability - HealthVerity

HealthVerity's tokenization/de-identification software utilizes a single CPU core for its operation and processed records at an average rate of approximately 6,285 records per second across all 10 source data files used in this evaluation.

HealthVerity hashes each identifier using a bloom filter and can hash all of the identifiers for a record in a single pass, which makes the tokenization/de-identification process very efficient.

Even though the HealthVerity tokenization software was very efficient, the fact that it is limited to a single CPU core means that it does not scale well. We would encourage HealthVerity to explore the possibility of upgrading its tokenization/de-identification software in the future to make use of multiple CPU cores. Such an enhancement may improve the already impressive processing speed significantly.

HealthVerity's tokenization/de-identification tool consistently used approximately 800 MB of memory regardless of which of the source data files were being de-identified.

We were unable to obtain any metrics regarding the performance and scalability of HealthVerity's matching utility because the matching took place behind HealthVerity's firewall.

Performance and Scalability – Datavant

When operating on a single CPU core, Datavant's tokenization/de-identification software processed an average of 1,443 records per second across all ten of the source data files. When operating on four CPU cores, the performance improved to an average of 4,037 records per second.

Our tests were conducted using a desktop PC with four CPU cores, but Datavant's tokenization/de-identification utility can also be run on a server or on a desktop PC with an increased number of CPU cores. The performance would likely improve considerably under these conditions, given that the software is designed to scale with the number of cores that are available, and it scales well.

Datavant's tokenization/de-identification utility generates multiple tokens per record. It does this by combining each of the identifiers from the source data in various ways and then hashing the result (for a full list of the tokens that were generated for this evaluation see Methods, Table 4). When data are hashed in this way, even the smallest differences between two values can produce radically different hash results. When this happens, records that belong to the same individual may not be linked. Datavant's tokenization software uses a probabilistic approach, where different combinations of identifiers are generated in tandem, to reduce the chance of a false negative result occurring. The logic behind this strategy is that the same individual will only differ on, at most, a small number of fields. Other methods for PPRL (such as bloom filters) have been shown to provide greater accuracy and efficiency but trade-off greater re-identification and security risks [2, 3].

Datavant's tokenization/de-identification software used approximately 50 MB when operating on a single CPU core and approximately 350 MB when operating on four CPU cores, meaning that the memory requirements scale with the number of CPU cores in use.

Datavant's matching tool utilizes a single CPU core for its work, and this number cannot be increased. The memory requirements of the matching tool increase with the number of records being matched and the number of tokens. In the case of the SEER linkage data (Scenario 1), 64 GB of memory were insufficient to perform the linkage in a single pass, and we needed to break the work up into three smaller linkages and combine the results to complete the linkage. While not a direct comparison, the linkage tool that was used to create the gold standard linkage results, Match*Pro, used less than 2 GB to link the raw data for the same number of records. We would recommend Datavant do whatever it can to reduce the memory consumption of its matching tool given what we observed in our testing.

Performance and Scalability – Senzing

Senzing's hashing software ran as a single-threaded process, not taking advantage of the computing power of the server, with CPU usage averaging around 4.6%. Multiple hashing instances were run simultaneously for different datasets to take better advantage of the server's computing power. The software hashed data at an average rate of 178 records per second across all ten of the source data files. Memory usage averaged around 14 GB.

Based on our recorded metrics it appears that the Senzing matching software could use the full computing power of the server as configured, with CPU usage averaging around 99%. The software matched data at an average rate of 311 records per second across each of the three scenarios we examined. Memory usage averaged around 14 GB.

Performance and Scalability – Veeva SafeMine

Veeva's hashing software ran as a single-threaded process, not taking advantage of the computing power of the server, with CPU usage averaging around 9.4%. We attempted to run multiple hashing instances simultaneously, but as mentioned before, the database server became bottlenecked after running just two concurrent hashing processes. The software hashed data at an average rate of 115 records per second across all ten of the source data files. Memory usage averaged around 19 GB.

We did not license the SafeMine matching software, so we did not collect any performance and scalability metrics for it. The matching process took place behind Veeva's firewall.

FINAL THOUGHTS

The process of conducting privacy-preserving record linkages is complex and the approaches taken to perform these linkages can vary significantly from one vendor to another. The results of scenarios 2 and 3 suggest that, regardless of the approach used, privacy-preserving record linkages are more negatively affected by typographical errors and lower levels of completeness than standard (non-privacy preserving) record linkages. We base that assertion on the fact that the F1 scores for scenario 3 were always significantly lower than the F1 scores for scenario 2 for the same software package and/or token set and the fact that hashing algorithms and bloom filters (though often to a lesser extent) are innately sensitive to minor discrepancies in the data.

Before using any PPRLS, potential users should consider the biases, strengths, and weaknesses inherent to each PPRLS (and PPRLS as a whole) as well as their own use case, the quality and completeness of their data, and their tolerance for false positive and false negative matches.

REFERENCES

1. Jesse Aronson et al. *Landscape analysis of Privacy Preserving Patient Record Linkage Software (P3RLS) Final Report*, prepared by Synectics for the National Cancer Institute, August 15, 2019.

2. Randall S, Wichmann H, Brown A, Boyd J., Eitelhuber T, Merchant A, Ferrante A. A blinded evaluation of privacy preserving record linkage with Bloom filters, *BMC Medical Research Methodology*, Jan 2022.

3. Randall S, Brown AP, Ferrante AM, Boyd JH. Privacy preserving linkage using multiple match-keys. *Int J Popul Data Sci*. 2019;4(1):1094.

ACKNOWLEDGEMENTS

Ryan Basile (Senzing)

Charlotte Bourg (Datavant)

Itzhack Erlich (Veeva)

Cree Flory (HealthVerity)

Whitney Kemper (Veeva)

Andrew Kress (HealthVerity)

Will Layton (Senzing)

Sandy Leonard (HealthVerity)

Paul Lohr (Senzing)

Jasmin Phua (Datavant)

Mark Ungerer (Datavant)