A New Algorithm to Efficiently Match U.S. Census Records and Balance Representativity with Match Quality

Eric Protzer Sultan Orazbayev Andres Gomez-Lievano
Matte Hartog Frank Neffke

November 21, 2024

Abstract

We introduce a record linkage algorithm that allows one to (1) efficiently match hundreds of millions of records based not just on demographic characteristics but also name similarity, (2) make statistical choices regarding the trade-off between match quality and representativity and (3) automatically generate a ground truth of true and false matches, suitable for training purposes, based on networked family relationships. Given the recent availability of hundreds of millions of digitized census records, this algorithm significantly reduces computational costs to researchers while allowing them to tailor their matching design towards their research question at hand (e.g. prioritizing external validity over match quality). Applied to U.S Census Records from 1850 to 1940, the algorithm produces two sets of matches, one designed for representativity and one designed to maximize the number of matched individuals. At the same level of accuracy as commonly used methods, the algorithm tends to have a higher level of representativity and a larger pool of matches. The algorithm also allows one to match harder-to-match groups with less bias (e.g. women whose names tend to change over time due to marriage).

1 Introduction

The recent digitization of hundreds of millions of historical census records, together with significant advances in computing resources, has led to a surge of economics research on long-term trends in economic mobility, inequality, and other social outcomes (Ruggles et al., 2024). A major effort in this area has been matching census records over time to allow for longitudinal research. The focus so far has mostly been on reducing type I errors (making as few false matches as possible) as well as type II errors (making as few missed matches as possible). Helgertz et al. (2022), for instance, match about 46% of a sample of 100,000 men across a U.S. Census wave, improving earlier efforts by almost 20 percentage points. Their type I and type II are respectively 68% and 65% lower than earlier best efforts.

A major challenge, however, is that matching quickly becomes computationally expensive with larger populations, with trillions of possible matches to consider for U.S. Census waves.

This prompts many authors to predefine matching restrictions such as matching only those who share the same first letter of the first name or those who live in similar households over time. A key question, in turn, is how to assess and maintain representativity and external validity in matched samples. Unobserved heterogeneity and selection bias among matched and unmatched individuals can lead to biased estimates, for instance, of the treatment effect in causal inference studies. For example, using household data to match individuals over time (e.g. living together over time with a father and mother having the same name) would yield significantly higher match rates for those individuals that live in stable households, at the expense of skewing matches against people who switch households.

We address these challenges in a number of ways. First, we introduce a computationally efficient algorithm that allows us to relax some of the strict conditions prevalent in the literature that are used to filter for candidate matches. Second, we introduce an approach that allows researchers to estimate and make statistical choices concerning the tradeoff between match accuracy, pool size, and representativity. Third, we introduce a method that allows one to algorithmically create a ground truth of true and false matches suitable for training purposes, based on networked family relationships. This method reduces the burden on researchers to create ground truths by hand. It may come in handy if new census records become available (such as the 1950 U.S. Census records, at the time of writing this paper), which can be readily matched without going through the costly process of creating hand-crafted ground truths again. Fourth, our algorithm allows one to include groups that are often underrepresented in matching efforts in a more representative way (for instance, women are much harder to match because of marriage inducing a last name change, greatly increasing the chance for unmarried women to be overrepresented in matched samples). Fifth, we aim to make the matched datasets on U.S. Census Records publicly available at different trade-off values between quality and representativity, allowing researchers to use them hands-on.

2 Record Linkage Matching Efforts and Challenges

As Ruggles et al. (2024) document, census matching efforts have been done mostly using U.S. records - some dating back to the first half of the 20th century (e.g. Malin et al., 1935; Owsley and Owsley, 1940). In the 1980s the first national samples of linked data were created (e.g. Guest, 1987; Steckel, 1988) but the quality in terms of sample size and false matches was generally low. All of these efforts were done by hand.

Automated linking efforts took off with the release in 2003 of the 1880 full-count U.S. Census data at the individual level by the Integrated Public Use Microdata Series (IPUMS - Roberts, 2003; Ruggles et al., 2003). Ferrie (2005) linked these data to 1% samples in earlier and later censuses, and IPUMS subsequently published the Linked Representative Samples to allow researchers to follow individuals over time and accordingly study changes in census variables (Goeken et al., 2003, 2011).

The efforts above were limited to small samples, with Abramitzky et al. (2012) being the first to implement automated matching for full census counts. This work - and other similar efforts (e.g. Abramitzky et al., 2014; Beach et al., 2016) - use a preset of rules to determine a match, relying primarily on individual factors that are fixed over time such as

name (phonetic classifications), year of birth and location of birth.

To further improve match quality, recent years have seen a surge in the use of machine learning (e.g Abramitzky et al., 2020; Price et al., 2021). This generally involves training a model on a subset of hand-checked high quality data that serves as ground truth, which is subsequently used to predict matches in the full data. Bailey et al. (2020), for instance, use human input to label true and false links between individual records over time. Price et al. (2021) use records on individuals' histories (obtained from platforms like Ancestry) that are much more detailed than what is available in census records as such, allowing for a much higher degree of accuracy in matching such individuals over time.

At the same time, recent efforts have started incorporating information on matches that goes beyond immutable characteristics at the individual level, particularly family and household information. For instance, an individual that lives in a household with other individuals with stable names across decades (e.g. the mother and father) has a much higher chance of being matched correctly. Correspondingly, Fu et al. (2014) find that incorporating a measure of household similarity increases accuracy at the individual level. Similar follow-up efforts use household-level information in different ways, from it being the primary source to identify individual links over time (Rijpma et al., 2020) to it being used in ground truth creation (Antonie et al., 2014) - all of which generally increase match quality.

A major challenge in these efforts, however, is computational efficiency. A common first step in matching is calculating name similarity. On modern computers, calculating a single Jaro-Winkler distance between two strings takes a few microseconds ($= 10^{-6}$ s). With hundreds of millions of individuals to match - in the case of, for instance, U.S. Census records - there are roughly 10^{16} calculations to perform. This would result in a total computation time of about $10^{16} \times 10^{-6} s \sim 10^{10} s$ - 317 computer years on a single core. Given census waves spanning multiple decades, even with vast supercomputing environments with thousands of cores nowadays available to researchers, this is still prohibitively expensive in terms of both time and cost.

As a result, strict pre-determined matching conditions (called "blocking") are often implemented. It is common, for instance, to match only those individuals who share the same first letter of the last name, the same birthplace, and very similar years of birth. Alternatively one may match only those that have similar features at higher levels of aggregation (e.g. at the household level, one may match only those sharing the same mother's and father's name).

A key challenge, however, then, is to maintain representativity in matched samples. Ruggles (2002) already argued that using information on family members, for instance, could introduce significant selection bias. Individuals in stable households, for instance, would be more likely to be matched over time, as well as those that do not move location. The same holds for matching on occupation and place of residence (with non-movers being more easily matched). Determining this trade-off between match quality and representativity highly depends on the research question at hand - particularly for causal inference and external validity, it is important to maintain representativity in matched samples. However, there is no algorithmic solution to this yet, with Bailey et al. (2020) noting, in their evaluation of existing record linkage algorithms, that "that no linking method produces samples that are consistently representative of the linkable population, and the ways in which the data are not representative differ by algorithm".

As a result, this issue so far has been dealt with in very specific contexts and for narrowly defined groups. Ward (2023), for instance, uses doubled-linked data samples and instruments second-father-observations with first-father-observations to address measurement error affecting intergenerational mobility rates. While such approaches are highly innovative they unfortunately greatly limit, due to data requirements, the scope of research questions at hand.

At the same time, in full-scale matching efforts, certain population groups tend to be underrepresented. Women, for instance, are much harder to match because of surname changes due to marriage. As a result, unmarried women tend to be overrepresented in matched samples.

To address these issues, we present a novel approach to match a large number of records which is computationally efficient and uses less strict conditions to filter for match candidates. At the same time, we present a more systematized approach to determine the tradeoff between match quality, pool size, and representativity. Our algorithm allows researchers to focus on either of these three dimensions in their record linking efforts, allowing them to make statistically-informed choices to justify their linking strategy. Using this algorithm, we showcase how women can be included in matched samples with greater representativity.

3 Matching Algorithm

Our matching algorithm takes in individual-level microdata in two time periods, and leverages individual-associated variables (such as name, birthplace, and birth year) to match people across the two waves of data. There are three major stages in our matching algorithm. Stage 1 generates both match candidates and a pool of ground truth and false matches, without using any external hand-linked matches. Stage 2 creates one version of matches that are designed to prioritize representativity. Stage 3 creates another version of matches that are designed to maximize the size of the match pool, expressly at the expense of representativity. Each of these stages are described below.

3.1 Stage 1

Figure 1 provides an overview of Stage 1 of our algorithm. At a high level, our algorithm begins – like many other approaches in the literature – by using blocking (filtering based on similar demographic variables) to identify plausible match candidates based on their shared characteristics. Our algorithm, however, introduces computationally-efficient name matching, which allows us to consider less strict blocking criteria on other demographic variables than other approaches in the literature. Furthermore, we then leverage family-level relationships between matched candidates to create a set of ground truth and false matches, which we use to train subsequent stages of the algorithm.

Many matching algorithms use relatively strict blocking, where potential match candidates are only considered if they match exactly or near-exactly on certain demographic criteria. Abramitzky et al. (2021), for example, variously require in their algorithms that candidates match on birthplace, have birth years at most three years apart, and match on first and last initials. While strict blocking dramatically reduces the number of match

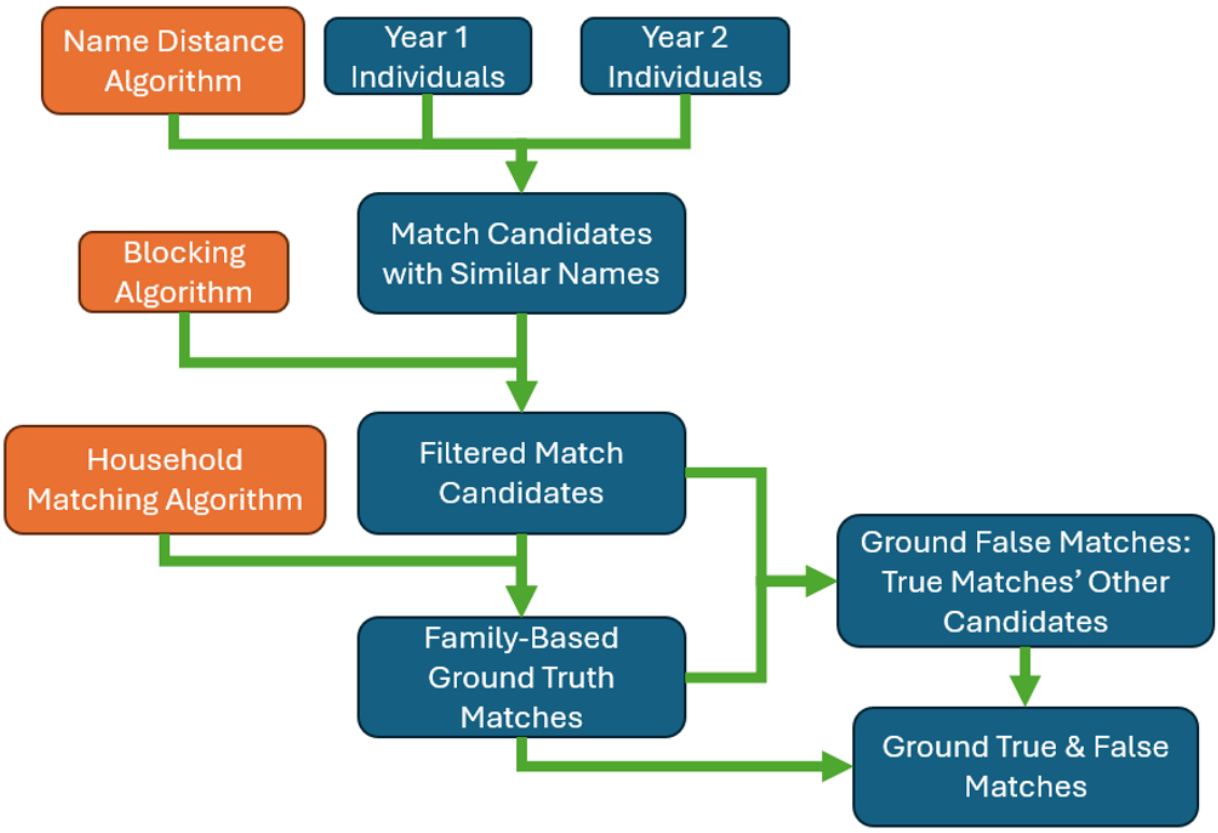


Figure 1: Stage 1 of Our Matching Algorithm

candidates an algorithm has to consider, it also discards potential matches who may have incurred transcription or other human errors. As discussed in Appendix C, we draw a subsample of hand-made genealogy matches from Family Tree that is representative of the potentially-matchable population, and calculate that approximately 8% of men in 1900 - 1910 are discarded by strict blocking.

In contrast, we are able to use less strict blocking criteria by introducing an algorithm that efficiently compares the name similarity of all possible pairs of individuals. Computing the name similarity through conventional metrics (for example, the Jaro-Winkler [JW] similarity) of all these possible pairs would take years of computing time.We design a more computationally efficient approach that, at a high level, uses the following steps:

1. Reduce the number of comparisons to *unique names* only, not individuals.

2. Represent names as a *numeric vector* of coordinates, rather than as character strings.

3. Use a state-of-the-art approximate nearest neighbor (ANN) algorithm.

The first step significantly reduces computation time by measuring the distance between unique names rather than individuals, since names are often repeated across different people. Thus, rather than comparisons between hundreds of millions of individuals, we compare

millions of names, reducing the number of pairwise computations be several orders of magnitude.[1] The second step allows for more efficient use of computational resources and enables the implementation of the third step, which leverages advances in numerical matching algorithms.

After completing steps 1–3 and finding close matches for each unique name, we need to transfer these matches to the individual level. This transfer can be achieved through a data merge operation (assuming the individual-pair data are stored in long format) or through matrix multiplication (if individual-pair data are stored in wide-format matrices). We opt for the latter, as these matrices contain many zeros and can be efficiently represented using sparse matrix techniques.

The mathematical operation to go from matches between names to matches between individuals is as follows. Let $\mathbf{S}_{12}$ be the $u_1 \times u_2$ matrix that compares u_1 unique names in year 1 with u_2 unique names in year 2 (this matrx can be densely populated with continuous numeric similarities between names, or can be already a matrix of binary values indicating the best matches). Next, let $\mathbf{I}_1$ and $\mathbf{I}_2$ be matrices of individuals in years 1 and 2 of dimensions $n_1 \times u_1$ and $n_2 \times u_2$, respectively, where n_1 and n_2 are the number of individuals. These $\mathbf{I}$ matrices are lookup tables where each row corresponds to an actual individual and has a single element with a 1 indicating the unique name assigned to that individual and zeros elsewhere. The matrix of matches between *individuals* is then given by the following matrix multiplication:

$$\mathbf{F}_{12} = \mathbf{I}_1 \cdot \mathbf{S}_{12} \cdot \mathbf{I}_2^T. \tag{1}$$

This strategy, expressed through Equation (1), of taking matches between characteristics and then expanding or transferring them to actual individuals can be applied beyond names. Thus, in general, one can create matrices $\mathbf{S}_{12}$ for different types of variables k, corresponding to various characteristics such as first name, last name, sex, age, location, among others. If the type of matching between individuals that is being implemented requires that two individuals match *only* if they are similar across every characteristic, then an element-wise multiplication of the final matrices should applied:

$$\mathbf{F}_{12} = \coprod_k \mathbf{F}_{12}(k) = \coprod_k \mathbf{I}_1(k) \cdot \mathbf{S}_{12}(k) \cdot \mathbf{I}_2(k)^T, \tag{2}$$

where $\coprod$ represents element-wise multiplication across characteristics k of the elements of the matrices that are multiplied. In this study, we implement this approach for $k = \{\text{first names}, \text{last names}\}$.

While computing the similarity of unique names rather than individuals is more efficient, it remains computationally expensive because there are still millions of unique first and last names. To address this challenge, we estimate the matrices $\mathbf{S}_{12}$ not by directly computing JW distances between all pairs of unique names—which is computationally intensive—but by converting names into numeric vectors and computing the Euclidean distance between them, which is much more efficient. We draw inspiration from the concept of triangulation (or more precisely, trilateration). The key idea is that the similarity between two names

[1]The distributions of name frequencies are well-known to have very long tails, with the majority of individuals represented by very few names.

can be inferred from each name's JW distances to a small set of shared reference names, or "beacons". In this approach, each name is represented by a vector of its JW distances to these few beacons. Names that are similar will have similar distance vectors because they are at comparable "distances" from the same set of reference names. By computing the Euclidean distance between these vectors, we can efficiently approximate the similarity between names without computing all pairwise JW distances (see Appendix A for an example).

As has been emphasized above, computing Euclidean distances is orders of magnitude more efficient than computing JW distances. As a consequence, in large data sets, this reference-based procedure generates remarkable computational savings. To see this, assume that the number of unique names in year y_1 is about the same size as the number of unique names in year y_2, $u_1 = u_2 = u$. A brute-force exhaustive calculation of *all* JW pairwise distances requires $c_{all} \approx (1/2)u^2$ computations. In contrast, assuming d references, the reference-based procedure requires the computation of $u \times d$ JW distances for the first year's dataset and $u \times d$ for the second year. In total, $c_{ref-based} = 2du + (\varepsilon/2)u^2$ JW computations, where the second term is the calculation of Euclidean distances, and ε denotes the relative computational cost of Euclidean distance relative to a JW score. For the reference-based approach to be better than the brute-force approach, then d must be small. Specifically, $d < (1 - \varepsilon)u/4$. Since $\varepsilon \approx 0.1$ (or less) and u is on the order of millions, d can be up to a thousand while still achieving computational savings. In our algorithm for matching U.S. Census data, we used $d = 400$ reference names to represent all first and last names. For any given year y_i, the output of this process is a matrix $\mathbf{R}_{y_i}$ of dimensions $u_i \times d$ (i.e., representing names as row vectors of JW scores to the reference names).

In the final step, the Euclidean distances are computed with a state-of-the-art nearest-neighbors algorithm. Aumüller et al. (2020) recently provided performance benchmarks of methods in data sets of different sizes and dimensionality, specifically called *approximate* nearest neighbors (ANN), which allow comparing different approaches for finding true nearest neighbors to one another . The word 'approximate' here means that ANN methods can trade off speed against accuracy. The performance metrics used in these methods are "query time" (number of queries per second) and "recall" (or true positive rate). According to Aumüller et al. (2020), the best performing method at the time of their analysis was the Hierarchical Navigable Small World algorithm (HNSW) (Malkov and Yashunin, 2018). We use this algorithm to create k-nearest-neighbors ($k$nn) for each unique name and we populate the matrix $\mathbf{S}_{12}$ accordingly.

Having identified pairs of individuals with reasonably similar names, we use another algorithm to impose further blocking constraints (see Algorithm 1). First, we filter for those pairs with matching gender. Secondly, we further restrict these based on similarity in first name, last name, birth place and birth year, but with some flexibility to allow us to identify matches who may have some transcription error. We obtain our final set of filtered match candidates as a result.

We next leverage these candidates' family relationships (i.e. household membership) to generate a set of ground truth and false matches. The basic underlying logic is that when multiple match candidates appear in the same household in both waves of a census, it is very likely that they are true matches. Trying to match a John Smith to another John Smith in the next census wave may be very difficult, but the match is significantly more certain if one observes John Smith living with his wife Abigail Smith and his son Benjamin Smith in

Algorithm 1 Filter Match Candidates with Similarity Measures

```
procedure FilterCandidatesWithSimilarityMeasures(List of individual pairs)
    Initialize an empty list: plausibleMatchCandidates
    for each pair (individual1, individual2) in the list of individual pairs do
        if individual1.gender == individual2.gender then
            Compute JaroWinklerScore for first names: firstNameScore
            Compute JaroWinklerScore for last names: lastNameScore
            Compute birthplace match: birthplaceMatch (1 if match, 0 otherwise)
            Compute birth year proximity: birthYearProximity (1 if ≤ 3 years apart, 0 otherwise)
                                    ▷ First filtering criteria for transcription errors in demographics
            if (birthplaceMatch == 1 OR birthYearProximity == 1) AND (firstNameScore >0.9) AND (lastNameScore >0.9) then
                Add pair to plausibleMatchCandidates
            end if
                                        ▷ Second filtering criteria for transcription errors in names
            if (firstNameScore >0.9 OR lastNameScore >0.9) AND (firstNameScore >0.7) AND (lastNameScore >0.7) AND (birthplaceMatch == 1) AND (birthYearProximity == 1) then
                Add pair to plausibleMatchCandidates
            end if
        end if
    end for
    return plausibleMatchCandidates
end procedure
```

both waves. We require that households have at least four matching members in common, and that they only have that many matches in common with each other as opposed to any other possible households. See Appendix D for technical details. Helgertz et al. (2022) also leverage family information in their matching procedure, albeit to identify final matches as opposed to a ground truth set of matches for training purposes alone.

We take the set of match candidate individuals who are in matching households, and who have a unique match within that pair of households[2], to be our set of ground truth matches. We then take these individuals' other candidate matches (who are not in matching households) to be our set of ground false matches.

3.2 Stage 2

Armed with the pool of match candidate plus ground truth and false matches from Stage 1, Stage 2 trains machine learning algorithms and uses them to create matches designed to have good representativity. Figure 2 outlines the main steps of Stage 2. It begins by calculating a measure of individual-to-individual match quality based on demographic variables. It then compares this metric to each candidate's top three match quality scores, which helps to weigh if a match candidate is not only individually plausible but also relatively unique versus other candidates. Finally, it selects unique matches for each individual through a greedy selection algorithm that keeps candidate matches with the highest possible scores.

We load in a sample of ground truth and false matches, and train an xgboost machine learning algorithm to identify true matches based on gender, age, and similarity of first and last names, birthplace, and birth year. See Appendix E for technical details. We apply this algorithm to predict the probability that each possible pair of match candidates is true.

However, this metric by itself only measures the demographic similarity of two individuals. In reality the probability that a pair of match candidates forms a true match depends not just on how similar those two individuals are to each other, but also if they have other highly-similar matches. For example, a John Smith born in 1880 in Virginia whom we observe in 1900 might be very similar to a John Smith born in 1881 in Virginia whom we observe in 1910. But if there are multiple other John Smiths born around 1880 in Virginia, we cannot be so confident that the pair in question is a true match.

We thus train an additional, "near-neighbor aware" xgboost machine learning algorithm[3] that takes the top three match quality scores of each candidate into account. That is, if we consider a potential match between ID 1 and ID 2, we get the top three match scores for ID 1 and the top three match scores for ID 2. We include this information in the input data, along with the match quality score directly between ID 1 and ID 2. The form of the input data is denoted in Table 1.

We then apply this second xgboost algorithm to all possible match candidates to obtain a refined probability that each match is true. Finally, we need to ensure that each individual only has one selected match. We filter for candidates with a match probability of at least 0.5 (from the second, near-neighbor aware xgboost algorithm), then apply a greedy algorithm that selects the best possible set of matches.

[2]This avoids, for example, cases where two brothers in the same household might be difficult to distinguish.

[3]We tune hyperparameters for this algorithm in the same way as the previous xgboost algorithm described in the Appendix.

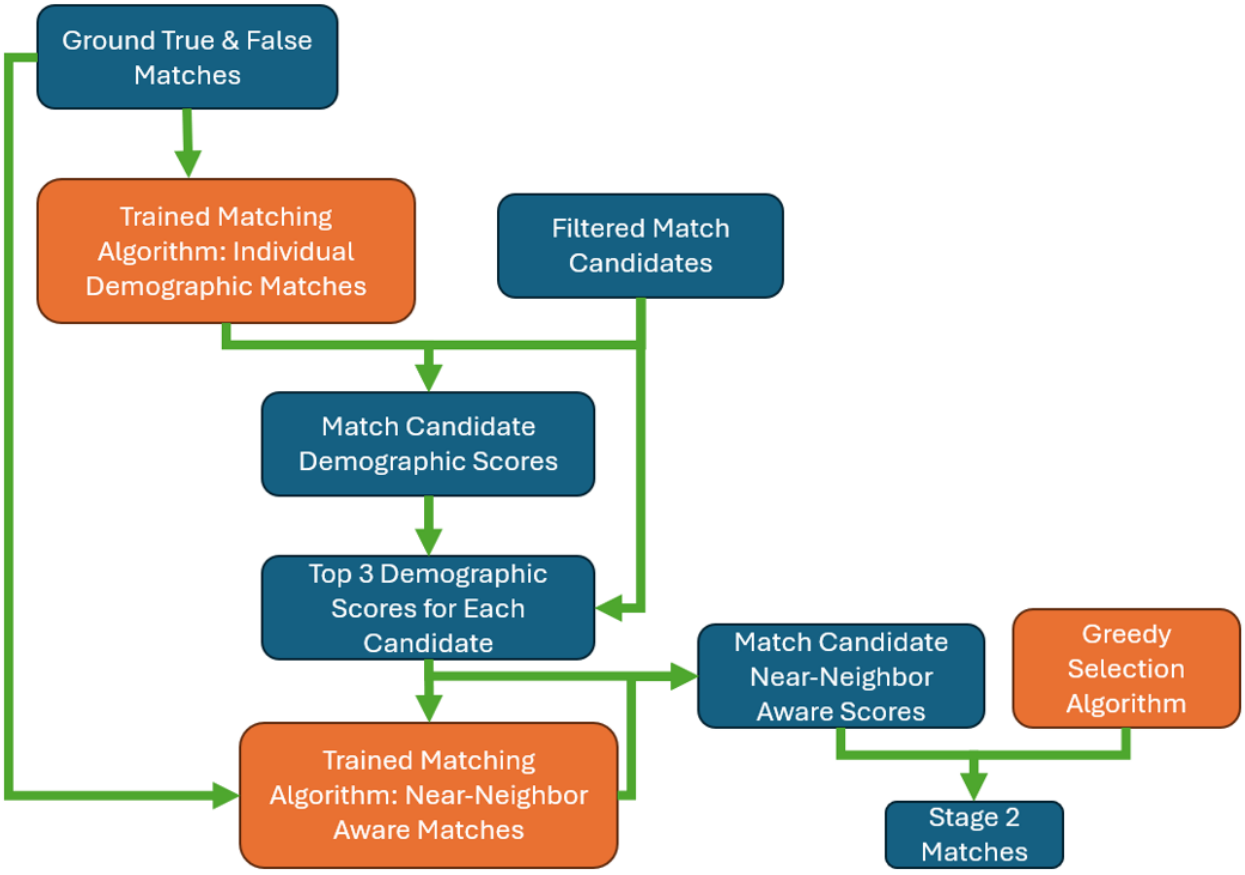


Figure 2: Stage 2 of Our Matching Algorithm

Table 1: Example Input Data for Near-Neighbor Aware Matching Procedure

Match Score for ID 1 and ID 2	**ID 1 First-Best Match Candidate Score**	**ID 1 Second-Best Match Candidate Score**	**ID 1 Third-Best Match Candidate Score**	**ID 2 First-Best Match Candidate Score**	**ID 2 Second-Best Match Candidate Score**	**ID 2 Third-Best Match Candidate Score**
0.8	0.9	0.8	0.5	0.8	0.75	0.4

The greedy selection algorithm sorts matches from highest to lowest probability score. It then iterates over the following steps:

1. Take the match with the highest score, and identify the IDs of the individuals in each wave (let us call these "ID A" and "ID B")
2. Add this match to an output file
3. From the pool of remaining matches, eliminate all matches that contain either ID A or ID B

4. Exit loop if the remaining pool of matches is empty

The resultant output file contains the set of 'best' matches, along with their associated match probabilities. We call these our "Stage 2 Matches." Within this set of matches one can filter for a minimum level of match probability to adjust the tradeoff between the size of the match pool versus match accuracy.

3.3 Stage 3

Our Stage 2 Matches are generated by matching on variables that are constant over a person's lifetime, such as name, birthplace, and year of birth. However, some approaches in the literature (such as Buckles et al. (2023)) acknowledge that a larger pool of matches can be obtained by leveraging variables that can change over a person's life, including location, household composition, and occupation, albeit at the expense of representativity (for instance, conditioning matches on being in the same county would skew the results against people who move). The third stage of our algorithm recognizes this, and produces an alternative set of matches that prioritize match pool size above representativity.

Figure 3 provides an overview of Stage 3 of our algorithm. We extract very high-quality working-age male matches from our Stage 2 matches, and fit a Logit model to estimate their probability of occupation matching based on age, occupation category, and urban vs rural status. We predict this probability for all working-age male match candidates, and subtract it from a binary indicator of whether each individual's occupation actually matches. We then fit an OLS regression that relates match quality, geographic information, and household information to this difference, the argument being that true matches are less likely to switch occupations than the average rate of their demographic group. We use the fitted regression to predict switching likelihood for all potential matches, which yields a final score of match quality that is adjusted for geographic and family similarity. We run this metric through a greedy selection algorithm to obtain final matches.

From the Stage 2 matches, we extract male matches age 20 – 54 in the first wave with a predicted true match probability of 0.99 or above. We focus on working-age men because they are likely to be employed, and we want to examine occupation match rates. We obtain or compute the following variables:

- A binary indicator for whether occupation matches across waves
- A binary indicator of urban versus rural status in the first wave
- Indicator variables for five-year age buckets, based on age in the first wave
- Indicator variables for aggregated occupation categories in the OCC1950 classification system in the first wave

We then run the following logistic regression:

$$OccMatch_{i,j} = \beta_0 + \beta_1 Urban_i + \sum_k \beta_k AgeBucket_{k,i} + \sum_l OccCategory_{l,i} + \epsilon_{i,j} \quad (3)$$

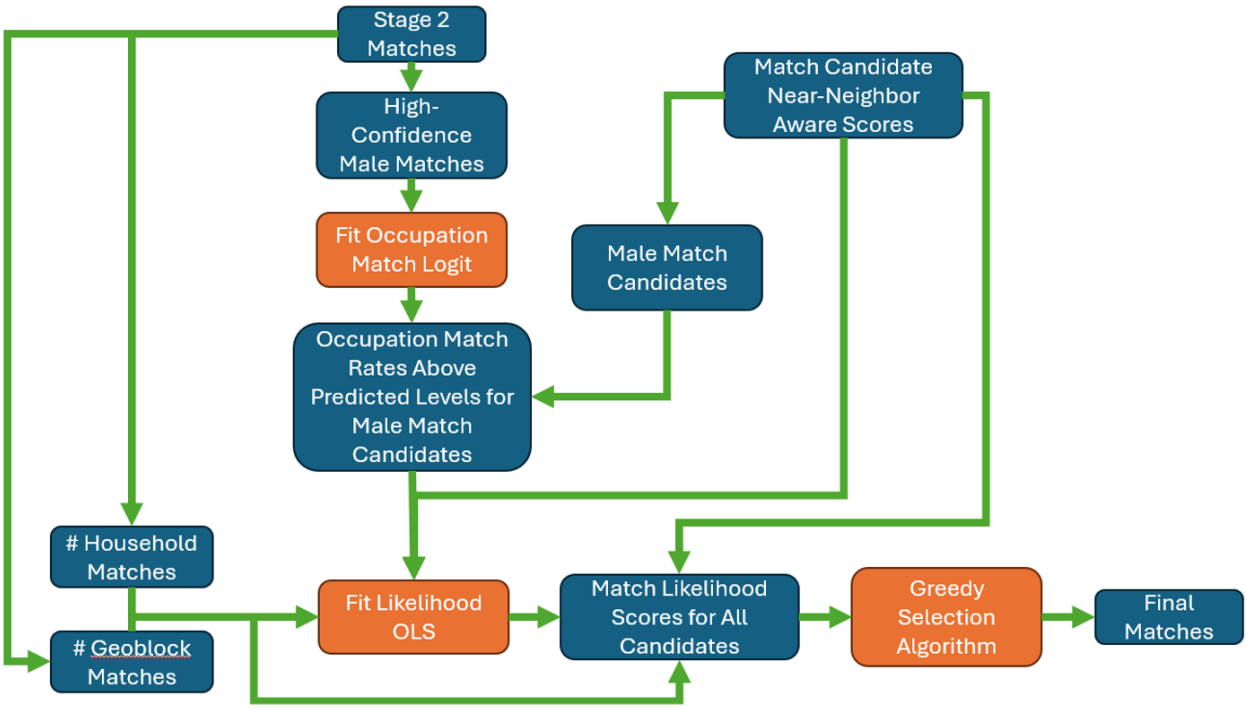


Figure 3: Stage 3 of Our Matching Algorithm

Having fitted the regression, we use it to predict occupation match rates for all working-age men from the pool of match candidates. We subtract the predicted match rate from a binary indicator of actual occupation matching. This differential is indicative of occupation matching at a rate above levels associated with a particular demographic bucket, which is informative because true matches should be likelier to remain in the same occupation than average.

Next we relate the following variables to this match rate differential, for all working-age male match candidates:

- The near-neighbor aware match probabilities from Stage 2 of our algorithm, which reflects demographic similarity and uniqueness.
- The number of Stage 2 matches shared by each match candidates' household. When this measure is high, the two households are likely to match.
- The number of "geoblock" Stage 2 matches between the candidates. A geoblock here is a group of individuals who were organized to be surveyed in sequence in the U.S. census, roughly reflecting a neighborhood. When the number of Stage 2 matches between two geoblocks is high they are likely to constitute a near-identical neighborhood. This metric thus reflects highly-granular geographic matching. See Appendix B for more technical detail.

Each of these variables has a non-linear relationship with the occupation match rate differential. We thus programmatically identify the knee in these relationships so that we can run a segmented OLS regression. To identify knees, we execute the following algorithm:

1. Take an input variable, sort it from smallest to largest value, and create 100 buckets that each contain an equal number of observations.

2. Within each such bucket, calculate 1) the mean value of $OccMatch_{i,j} - \widehat{OccMatch}_{i,j}$ and 2) the mean value of the input variable

3. Take these mean rates within each bucket as observations, and execute the kneedle algorithm (Satopaa et al. (2011))

Let $knee_{var}$ be the identified knee for each of the three variables. We can then create variables for the purposes of running segmented regressions; let $dummy_{var}$ be a binary variable that is 1 when the variable value exceeds $knee_{var}$, and $\overline{var} = (var - knee_{var}) \times dummy_{var}$. We then fit a segmented OLS regression as follows:

$$\begin{aligned} OccMatch_{i,j} - \widehat{OccMatch}_{i,j} = \beta_0 + \beta_1 MatchProb_{i,j} + \beta_2 \overline{MatchProb_{i,j}} + \beta_3 FamilyCount_{i,j} \\ + \beta_4 \overline{FamilyCount_{i,j}} + \beta_5 GeoCount_{i,j} + \beta_6 \overline{GeoCount_{i,j}} + \epsilon_{i,j} \quad (4) \end{aligned}$$

With this fitted regression, it is now possible to relate the three above input variables to an outcome that is indicative of match accuracy. We can thus run it to predict match accuracy for all match candidates. Having done so, we use the same greedy selection algorithm as described in Stage 2 to produce a set of "Final Matches." These Final Matches are, by design, less representative than our Stage 2 Matches, but offer larger match pools at a given level of match accuracy.

4 Performance of U.S. Census Matches, 1900 – 1910

We execute our algorithm to link all adjacent U.S. census waves from 1850 to 1940[4]. Next we need to compare the performance of our matches against other prominent examples in the literature. We evaluate our matches' performance against Abramitzky et al. (2021), who published the popular Census Linking Project (CLP), and Buckles et al. (2023), who published the Census Tree project that covers a very large pool of matches.

We evaluate our matches against other sources on several key dimensions: match accuracy, match pool size, and match representativity. An ideal set of matches would be highly accurate, in terms of containing very few false positives; large in terms of coverage, so as to maximize the statistical power of any analytical exercises the matches are used for; and representative of the general population, to ensure that any analytics that employ the matches are minimally biased. In practice, there are tradeoffs between these variables. A less strict set of criteria to identify a match can, for instance, increase the size of the pool of matches at the expense of match accuracy.

A ground truth dataset of some kind is necessary to evaluate how matches perform along our desired metrics. We consider the Family Tree dataset of hand-linked genealogy matches

[4]Like others in the literature, we are unable to link the year 1890 because the original census manuscript was lost in a fire. We thus link 1880 to 1900 to skip over 1890.

for the 1900 – 1910 U.S. census waves. Manually-constructed genealogy matches are often considered to be the "gold standard" in terms of match accuracy (see e.g. Bailey et al. (2020)), and the Family Tree dataset offers millions of such linkages between 1900 and 1910.

To compute match accuracy we filter for the set of individuals from a match source that appear at all in the Family Tree dataset, and calculate the share of these matches that agree with the Family Tree linkage. To evaluate the size of the match pool, we divide the total number of matches from a source by the number of potentially-matchable individuals. We define the latter to be those who appear in the second census wave being matched (and thus survived until at least that wave), were born no later the first wave being matched (and thus were alive at the time of that wave), and, for immigrants, who immigrated to the US no later than the first wave being matched.

Representativity is somewhat more involved to evaluate. Numerous papers apply "balance tests" that examine whether matched individuals statistically differ from the general population in terms of demographic variables such as race, gender, and age. We also, however, want to consider two dynamic factors: rates of county relocation and occupation switching. We calculate these rates from the more-representative subsample of the Family Tree genealogy matches that was previously mentioned.

First, we evaluate the tradeoff between match accuracy, match pool size, and representativity in terms of the share of people living in their state of birth in 1910. For this exercise we only examine US-born men, as comparing a person's state of birth versus residence is only relevant for those born in the US and Abramitzky et al. (2021) only cover men in their matches.

Both our Stage 2 and our Final Matches allow researchers to choose a desired minimum level of match quality to create a tradeoff between match pool size and accuracy. We thus consider several subsets of each of our types of matches, in each case filtered for a certain minimum level of match quality.

Figure 4 exhibits the results of this comparison. Our Stage 2 matches perform comparably or somewhat better than Abramitzky et al. (2021) in terms of representativeness at a certain level of match accuracy, and noticeably better than Buckles et al. (2023). The latter is likely more unrepresentative because it explicitly takes county-county distance among match candidates into account, which directly increases the match pool size at the expense of representativity. For a given level of match accuracy, our Stage 2 matches also offer larger match pools than Abramitzky et al. (2021). For instance, one set of CLP matches covers 28% of potentially-matchable US-born men at 90% match accuracy, while one set of our Stage 2 matches covers 37% of potential matches at 91% match accuracy. The latter pool of Stage 2 matches is thus nearly a third larger. Our Final Matches are often less representative, but offer a larger match pool size (albeit not as large as Census Tree).

Next we consider a different representativity variable: the rate of staying in the same county versus moving. Here we examine all men, and take the true rate to be that in the resampled Family Tree data. Figure 5 shows that our Stage 2 matches perform comparably or somewhat worse than CLP in terms of representativity at a given level of match accuracy, and again noticeably better than Census Tree matches. Stage 2 match pool sizes exceed those of CLP at a given level of match accuracy. Our Final Matches, once more, are less representative but cover a larger share of potential matches. For instance, at 87% match accuracy CLP covers 30% of potential matches; whereas at 86% match accuracy our Stage

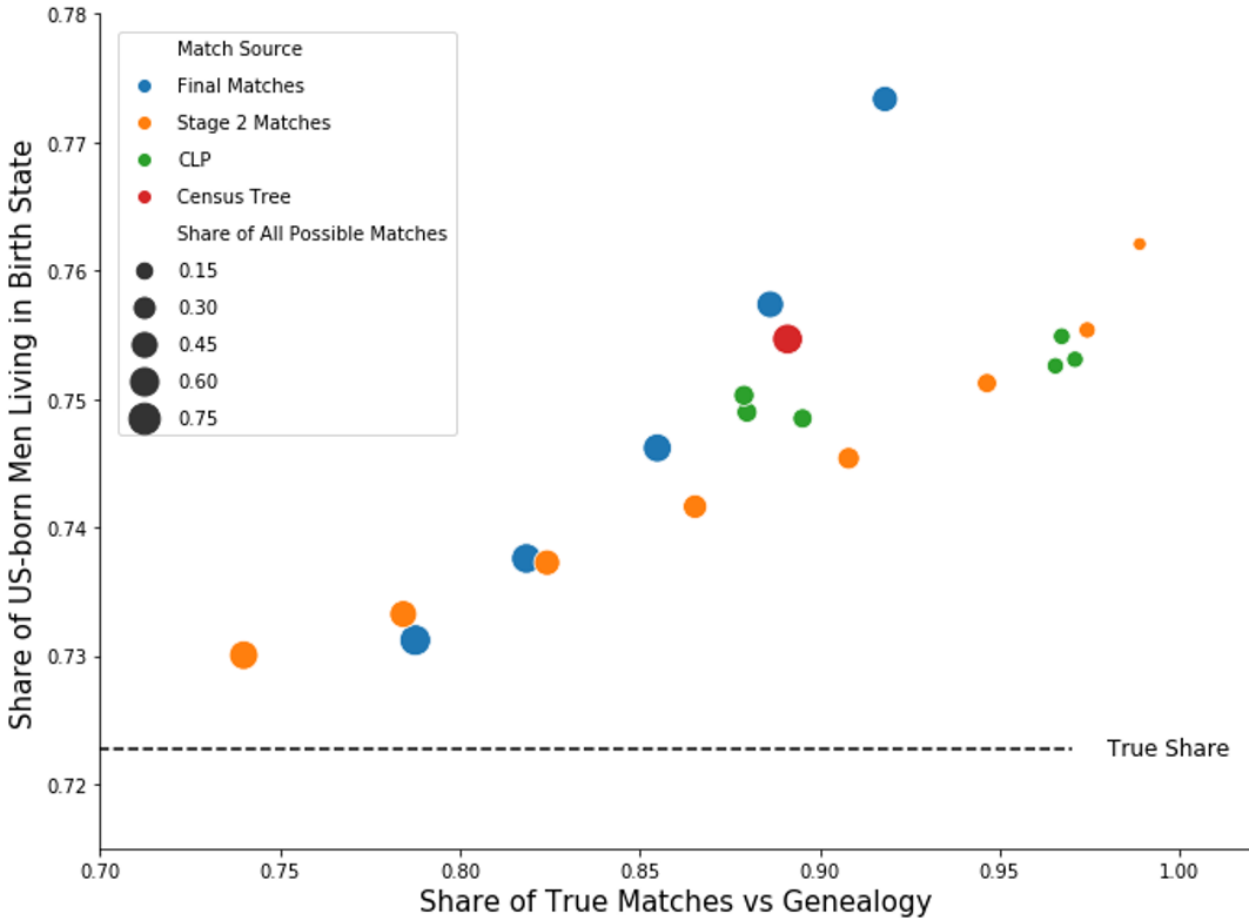


Figure 4: Match Comparison: Share of US-Born Living in Birth State

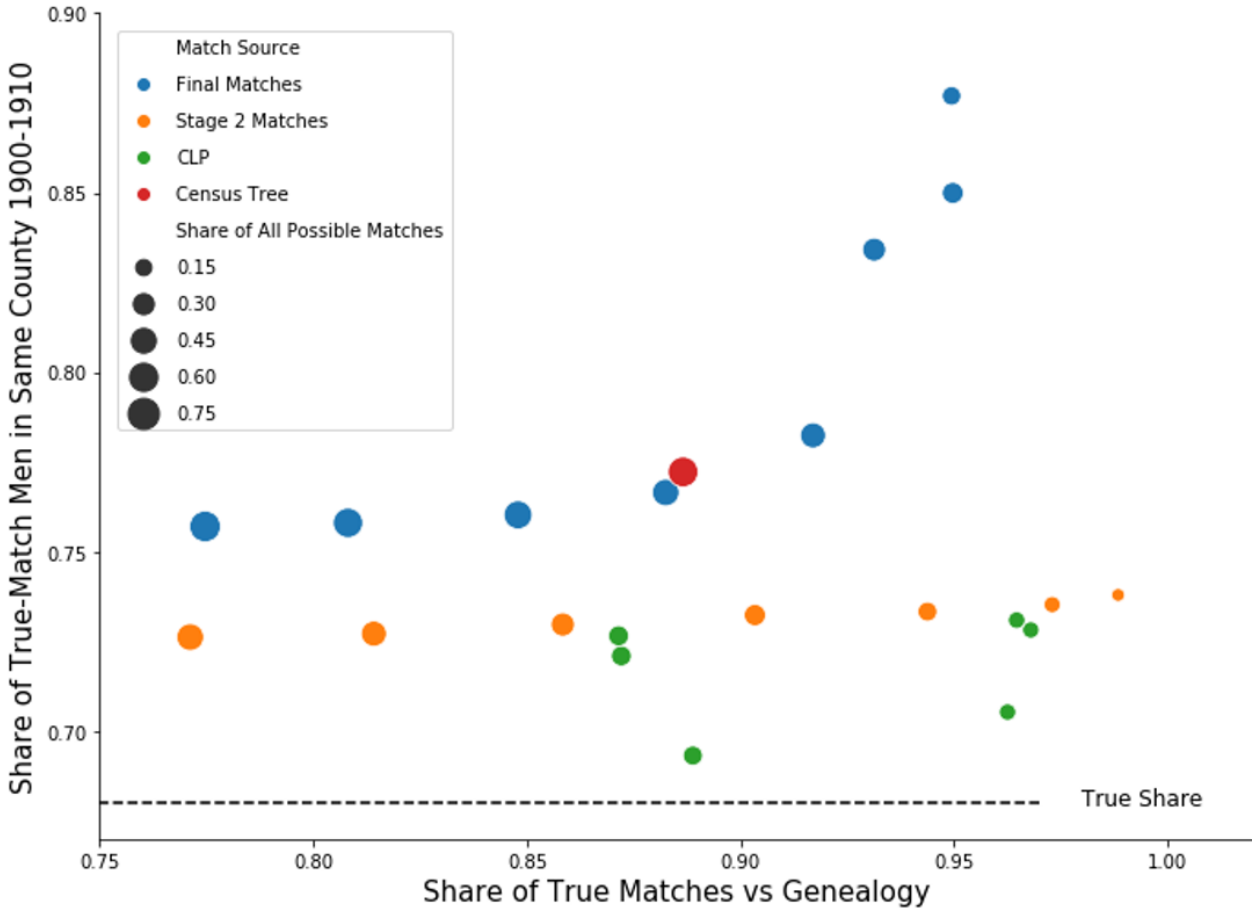


Figure 5: Match Comparison: Men in the Same County

2 matches cover 42% of potential matches, a 40% increase. Notably, all match sources show that men stay in the same county at plausibly high rates.

A further interesting feature of our Stage 2 matches' performance is that their representativity of county switching is highly stable with increasing match quality. In contrast, our Final Matches incur significantly increasing unrepresentativity after a certain point.

We also consider the rates at which prime-age men (those no younger than 20 in 1900 and no older than 55 in 1910) switch occupations. Critically, there was a major change in the way occupations were reported between 1900 and 1910, so we should not necessarily expect very high occupation match rates. Ward (2023) also convincingly argues that occupations were, in general, not accurately recorded in historical U.S. census waves, which would further reduce occupation match rates. Figure 6 shows that our Stage 2 matches perform comparably or slightly worse than CLP matches on representativity with regard to occupation switching, and again better than Census Tree. At 84% match accuracy CLP covers 26% of potential matches, while at 83% match accuracy our Stage 2 matches cover 38% of potential matches, which constitutes a 46% increase in the size of the match pool.

Another advantage of our matches is their coverage of particular demographic groups of interest. For instance, CLP matches do not cover women at all and while Census Tree does, the above results indicate that their approach sacrifices representativity. A major

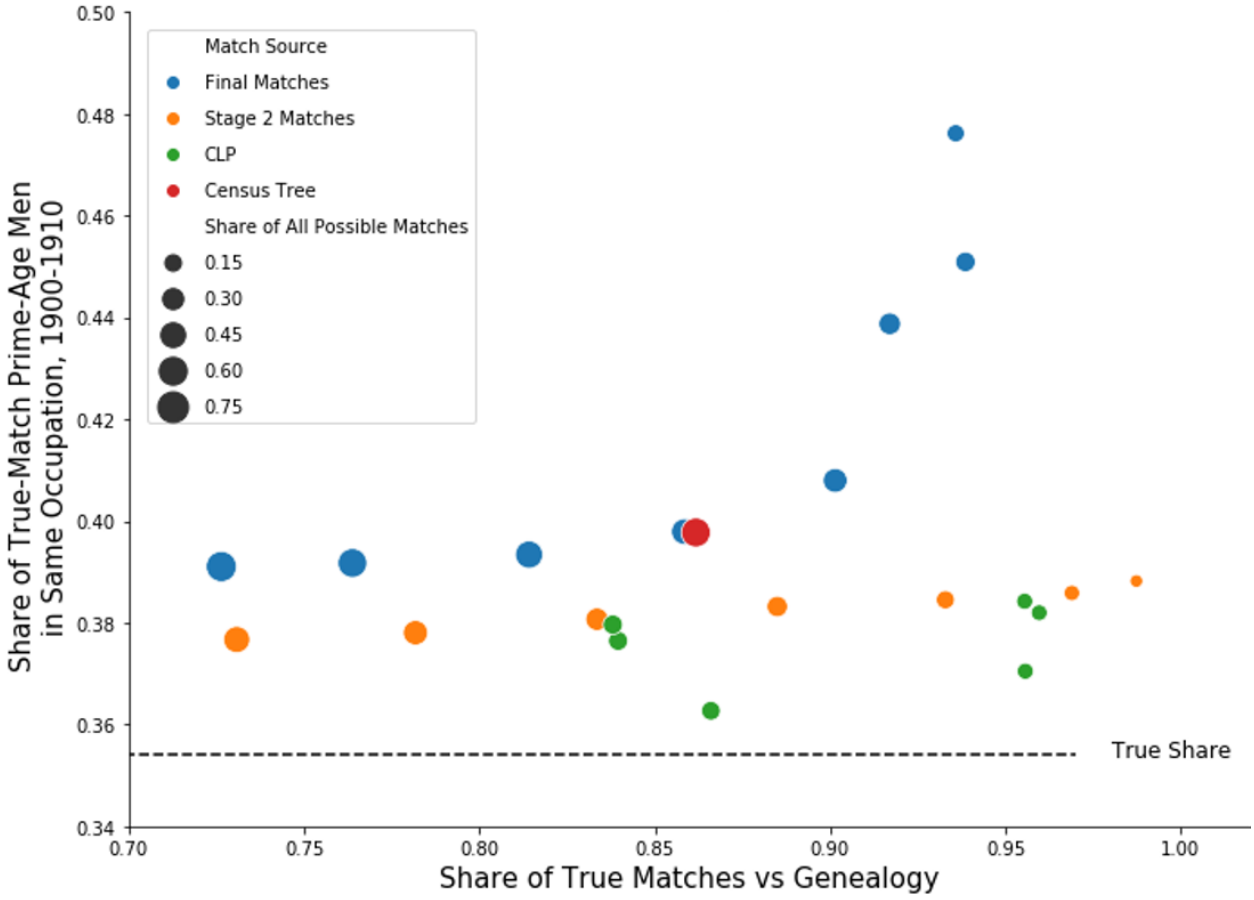


Figure 6: Match Comparison: Prime-Age Men in the Same Occupation

contribution of our work is thus a set of matches for historical US women, built in a way that does not incur explicit unrepresentativity. Figure 7 shows the tradeoff between the size of the match pool and match accuracy for women. In Appendix F we show that our Stage 2 matches for women are more representative than Census Tree with regard to the share of US-born women living in their state of birth.

Our approach also performs well on immigrants. Figure 8 shows that for foreign-born US men, our Stage 2 matches offer a comparable or larger match pool versus CLP matches. For instance, at 79% match accuracy CLP covers 19% of possible matches whereas at 78% match accuracy our Stage 2 matches cover 26% possible matches.

We additionally perform well matching African-Americans, as shown (for men only) in Figure 9. For instance, at 81% match accuracy CLP covers 13% of possible matches, while at 83% match accuracy our Stage 2 matches cover 17%.

In Appendix F we evaluate our matches versus alternatives (for men, to facilitate comparison with CLP) in terms of how representative they are of other key static demographic characteristics that are typically included in balance tests: age, geography, and race. Our matches generally perform well, and are often more representative than alternatives.

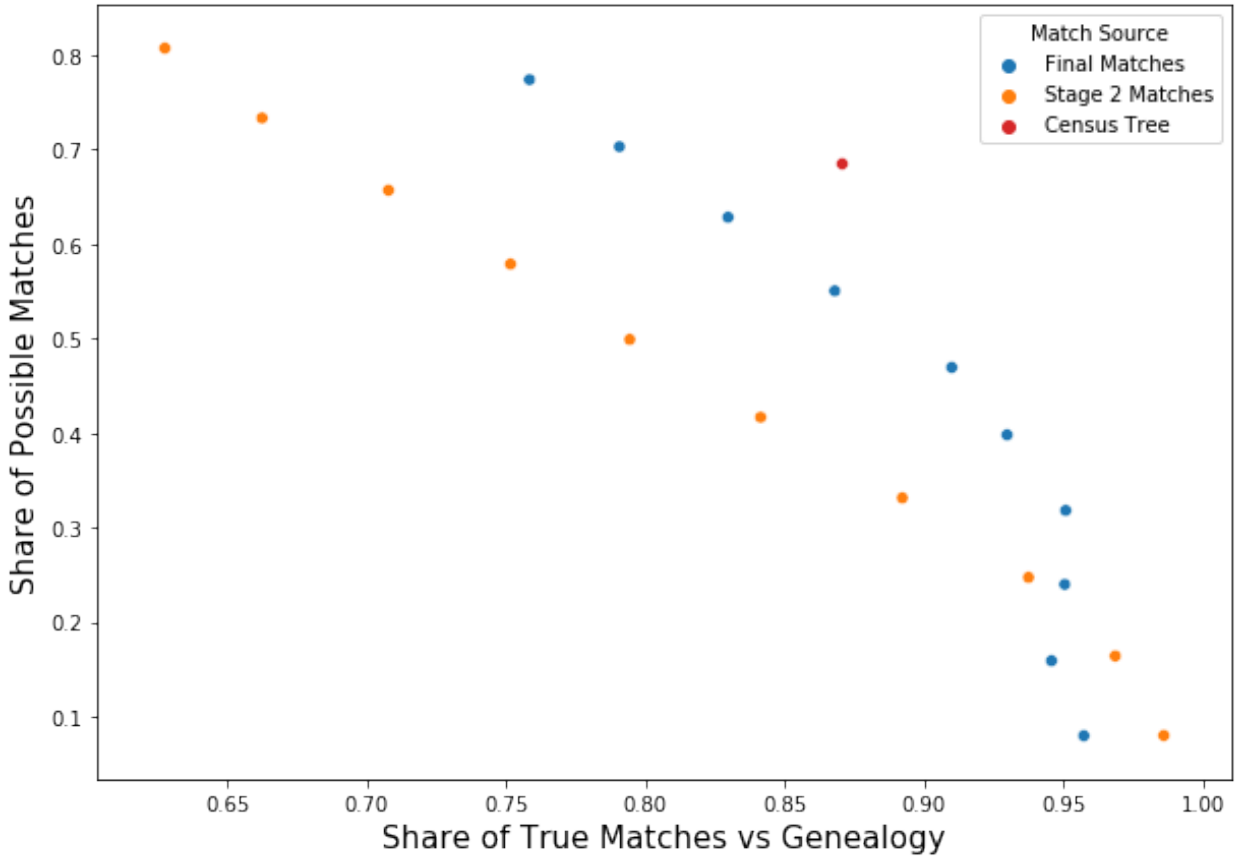


Figure 7: Match Comparison: Women

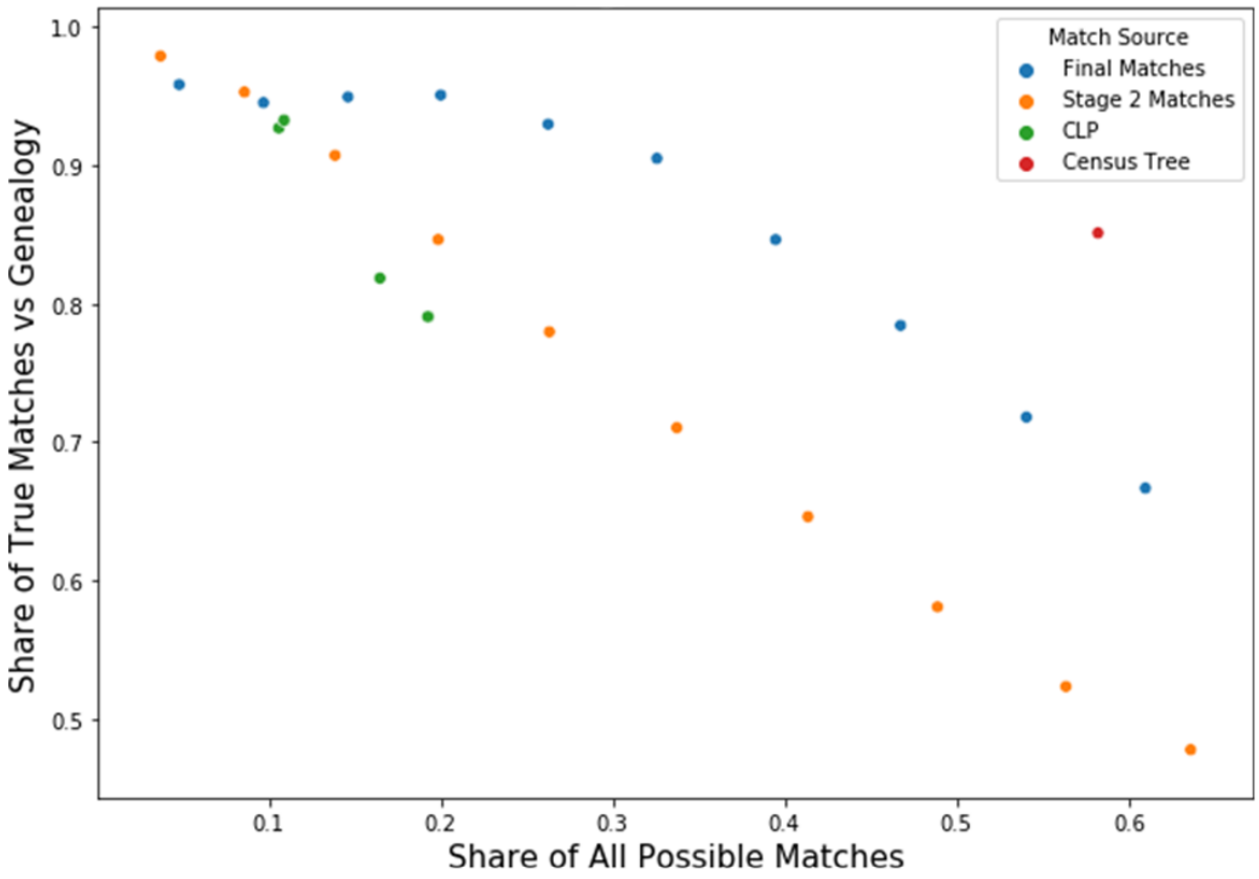


Figure 8: Match Comparison: Immigrants

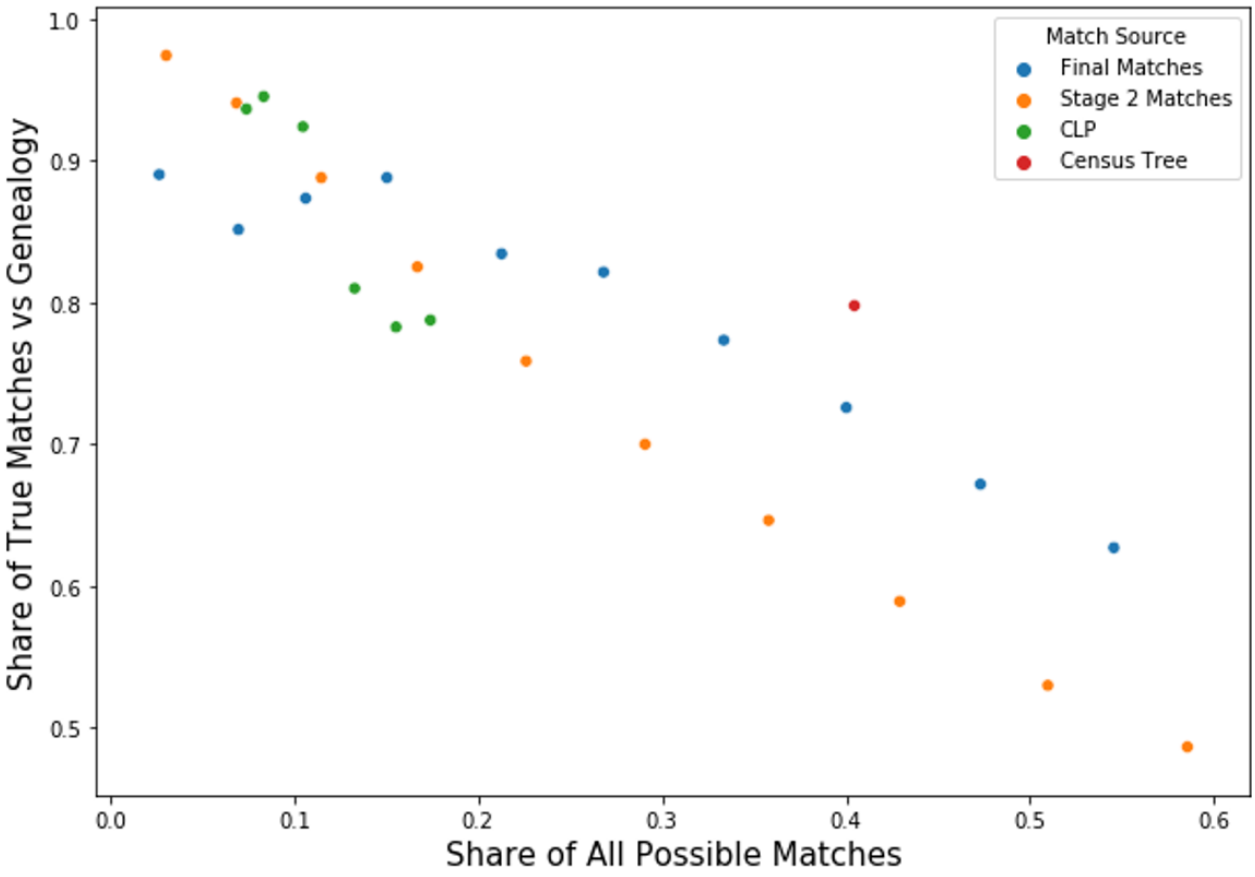


Figure 9: Match Comparison: African-Americans

5 Conclusion

We introduce a novel method of historical record linkage that leverages modern computational, machine learning, and network science techniques. Our methods introduce several significant innovations in the literature, and yield matches for the US historical censes that perform well versus alternative algorithms. Our matches are publicly available at: https://github.com/eric-protzer/census_ml_matches

Our approach to name distance measurement, for one, allows researchers to compare name similarity in a computationally efficient way. This introduces the possibility of more flexible use of name similarity as blocking criteria, which in turn enables researchers to loosen the strictness of other blocking criteria. Our experiments indicate that approximately 8% of potentially-matchable US men in the 1900 - 1910 wave are discarded by strict blocking criteria, so this innovation provides a way to reduce those losses.

We furthermore provide a way to create a ground truth dataset of matches for historical records, as long as those records contain information on which individuals are in the same household. Previously, machine learning methods in the literature such as Buckles et al. (2023) and Feigenbaum (2016) relied on hand-made matches of some kind, whether from trained research assistants or open-source genealogy platforms. Producing training data

computationally rather than by hand provides a major efficiency improvement that can be re-applied to other datasets and contexts.

The two-step method in which Stage 2 of our algorithm also advances neighbor-aware matching methods, where final match probability is not just informed by the quality of a particular pair of individuals but also the quality of their other candidate matches. Abramitzky et al. (2020) take a somewhat similar approach in which they require that each selected match's second-best match candidate should be below a certain quality threshold. Our approach is more flexible in that we can leverage a ground truth dataset to let the algorithm decide how much better the selected match needs to be than its near competitors.

As applied to historical U.S. censes, our algorithm also produces empirically useful matches. Our matches, in general, perform comparably or slightly better on representativity than Abramitzky et al. (2021)'s widely-used Census Linking Project while providing a larger pool of matches. They offer a smaller match pool than Buckles et al. (2023), but better representativity. While our matches only directly link adjacent censes, they can be adjoined together to link wider census intervals. Critically, we offer a sizable pool of matches for women that is more representative than the Census Tree project.

6 Acknowledgments

We would like to thank Ricardo Hausmann and the Growth Lab Fellows for the support, advice and suggestions that helped advance this project.

References

Steven Ruggles, Julia A Rivera Drew, Catherine A Fitch, J David Hacker, Jonas Helgertz, Matt A Nelson, Nesile Ozder, Matthew Sobek, and John Robert Warren. The ipums multigenerational longitudinal panel: Progress and prospects. *IPUMS Working Papers*, 2024.

Jonas Helgertz, Joseph Price, Jacob Wellington, Kelly Thompson, Steven Ruggles, and Catherine Fitch. A new strategy for linking us historicalc censuses: A case study for the ipums multigenerational longitudinal panel. *Historical Methods: A Journal of Quantitative and Interdisciplinary History*, 55(1):12–29, 2022.

James Claude Malin et al. turnover of farm population in kansas. *Kansas Historical Quarterly*, 4, 1935.

Frank L Owsley and Harriet C Owsley. The economic basis of society in the late ante-bellum south. *The Journal of Southern History*, 6(1):24–45, 1940.

Avery M Guest. Notes from the national panel study: Linkage and migration in the late nineteenth century. *Historical Methods: A Journal of Quantitative and Interdisciplinary History*, 20(2):63–77, 1987.

Richard H Steckel. Census matching and migration: A research strategy. *Historical Methods: A Journal of Quantitative and Interdisciplinary History*, 21(2):52–60, 1988.

Evan Roberts. Labor force participation by married women in the united states. In *Social Science History Association conference, Baltimore, MD, November*, volume 14, 2003.

Steven Ruggles, Miriam L King, Deborah Levison, Robert McCaa, and Matthew Sobek. Ipums-international. *Historical Methods: A Journal of Quantitative and Interdisciplinary History*, 36(2):60–65, 2003.

Joseph P Ferrie. History lessons: The end of american exceptionalism? mobility in the united states since 1850. *Journal of Economic Perspectives*, 19(3):199–215, 2005.

Ron Goeken, Cuong Nguyen, Steven Ruggles, and Walter Sargent. The 1880 us population database. *Historical Methods: A Journal of Quantitative and Interdisciplinary History*, 36 (1):27–34, 2003.

Ron Goeken, Lap Huynh, TA Lynch, and Rebecca Vick. New methods of census record linking. *Historical methods*, 44(1):7–14, 2011.

Ran Abramitzky, Leah Platt Boustan, and Katherine Eriksson. Europe's tired, poor, huddled masses: Self-selection and economic outcomes in the age of mass migration. *American Economic Review*, 102(5):1832–1856, 2012.

Ran Abramitzky, Leah Platt Boustan, and Katherine Eriksson. A nation of immigrants: Assimilation and economic outcomes in the age of mass migration. *Journal of Political Economy*, 122(3):467–506, 2014.

Brian Beach, Joseph Ferrie, Martin Saavedra, and Werner Troesken. Typhoid fever, water quality, and human capital formation. *The Journal of Economic History*, 76(1):41–75, 2016.

Ran Abramitzky, Roy Mill, and Santiago Pérez. Linking individuals across historical sources: A fully automated approach. *Historical Methods: A Journal of Quantitative and Interdisciplinary History*, 53(2):94–111, 2020.

Joseph Price, Kasey Buckles, Jacob Van Leeuwen, and Isaac Riley. Combining family history and machine learning to link historical records: The census tree data set. *Explorations in Economic History*, 80:101391, 2021.

Martha Bailey, Connor Cole, Morgan Henderson, and Catherine Massey. How well do automated linking methods perform? Lessons from US historical data. *Journal of economic literature*, 58(4):997–1044, 2020.

Zhichun Fu, Peter Christen, and Jun Zhou. A graph matching method for historical census household linkage. In *Advances in Knowledge Discovery and Data Mining: 18th Pacific-Asia Conference, PAKDD 2014, Tainan, Taiwan, May 13-16, 2014. Proceedings, Part I 18*, pages 485–496. Springer, 2014.

Auke Rijpma, Jeanne Cilliers, and Johan Fourie. Record linkage in the cape of good hope panel. *Historical Methods: A Journal of Quantitative and Interdisciplinary History*, 53 (2):112–129, 2020.

Luiza Antonie, Kris Inwood, Daniel J Lizotte, and J Andrew Ross. Tracking people over time in 19th century canada for longitudinal analysis. *Machine learning*, 95:129–146, 2014.

Steven Ruggles. Linking historical censuses: A new approach. *History and Computing*, 14 (1-2):213–224, 2002.

Zachary Ward. Intergenerational mobility in american history: Accounting for race and measurement error. *American Economic Review*, 113(12):3213–3248, 2023.

Ran Abramitzky, Leah Boustan, Katherine Eriksson, James Feigenbaum, and Santiago Pérez. Automated linking of historical data. *Journal of Economic Literature*, 59(3): 865–918, 2021.

Martin Aumüller, Erik Bernhardsson, and Alexander Faithfull. Ann-benchmarks: A benchmarking tool for approximate nearest neighbor algorithms. *Information Systems*, 87: 101374, 2020.

Yury A Malkov and Dmitry A Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

Kasey Buckles, Adrian Haws, Joseph Price, and Haley Wilbert. Breakthroughs in historical record linking using genealogy data: The census tree project. *National Bureau of Economic Research*, No. 31671, 2023.

Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. Finding a "kneedle" in a haystack: Detecting knee points in system behavior. *IEEE*, 31st international conference on distributed computing systems workshops:166–171, 2011.

James Feigenbaum. Automated census record linking: A machine learning approach. *Working Paper*, 2016.

7 Appendix

A Example of Reference-Based Matching of Names

Consider the example in Figure 10 of matching two data sets of unique names of sizes $u_1 = 5$ and $u_2 = 6$. Instead of computing $5 \times 6 = 30$ pairwise JW scores between all unique names in year 1 and all unique names in year 2, we have instead chosen two reference names, "Matte" and "Ljubica", and we have computed JW scores to those two names, for a total of $2 \times 10 + 2 \times 6 = 22$ comparisons.

The matching between these two data sets is created by computing the Euclidean distances between the coordinates of names, a computation that is much more efficient than the computation of string distances. Figure 11 shows the Euclidean distances for the example in Figure 10.

Census in **year 1**

	Matte	Ljubica
Andres	0.456	0.000
Sultan	0.456	0.540
Christine	0.437	0.418
Alexandra	0.374	0.418
Dario	0.467	0.448

Census in **year 2**

	Matte	Ljubica
Yang Li	0.448	0.429
James	0.600	0.000
Zultam	0.456	0.540
Andrez	0.456	0.000
Ulrich	0.000	0.540
Eric	0.000	0.595

Figure 10: Example of two lists of unique names taken from two censuses. Instead of computing all pairwise Jaro-Winkler similarities directly between the two lists, we have computed, for each year, Jaro-Winkler similarities only to two reference names.

	Yang Li	James	Zultam	Andrez	Ulrich	Eric
Andres	0.429	0.144	0.540	0.000	0.706	0.750
Sultan	0.111	0.559	0.000	0.540	0.456	0.459
Christine	0.015	0.449	0.123	0.418	0.454	0.472
Alexandra	0.074	0.475	0.147	0.426	0.393	0.414
Dario	0.027	0.467	0.093	0.448	0.476	0.490

Figure 11: Matches (indicated in red) between unique names in Figure 10 generated by computing Euclidean distances of the two-dimensional coordinates.

Similar names will have similar vectors and their mutual Euclidean distance will be short. In the example, Sultan and Zultam are represented by the same coordinates and are indeed very similar strings. It is worth noting that the converse is not always true; names with similar coordinates are not necessarily similar. This is the case for Christine and Yang Li, as can be seen in Fig. 11. However, the more reference names there are, the more accurately the vector distances will reflect the actual string distances between names. For example, we have found that 100 randomly chosen reference names suffices to disambiguate tens of thousands of different names. In future research, one could explore how many reference points are required to completely disambiguate every name in the data, as well as determining which names serve as the best reference points.

B Geocoding and Geoblocking

To geolocate individuals in census records, we use information on the number of the individual's census enumeration district, which is available from 1880 onwards. This complements information on the state, county, and city of people, grouping households within these geographical units into spatially contiguous groups. We link these numbers to descriptions of enumeration districts manually collected from Census forms by *stevemorse.org*, township data from Wikipedia, and information on precincts, beats and civil districts from the Compendium of the 1880 Census. We use optical character recognition and natural language

processing techniques to extract place information, which we in turn link to the Geographic Names Information System database (GNIS), as well as OpenStreetMap, Microsoft's Bing Maps, and Mapbox to obtain latitudes and longitudes for each place.

We additionally create geoblocks based on the way in which the Census data was collected and stored. Census enumerators were assigned specific geographic areas to cover (enumeration districts) of a size that a single enumerator could cover within a certain time. Subsequently, within each enumeration district, the enumerators would visit households in such a way to cover the area efficiently, moving from one household to those most nearby, which in turn is recorded by the enumerators (by hand) in the census schedules. Each household is recorded on a new line of the schedule, and as such, each page of the schedule contains multiple households. Upon completing the data collection, the collected data was transferred to microfilm for long-term storage (and easier access). All of this is recorded in the following IPUMS variables:

- **Reel Number (`reelno`)**: Each reel of microfilm contains a series of images from a range of enumeration districts.
- **Page Number (`pageno`)**: The page number corresponding to a specific page within the reel of microfilm.
- **Line Number (`lineno`)**: Each line on the census schedule corresponding to the respective household or individual.

Relatedly, the Census records contain, in addition to state information (`stateicp`) and county information (`county`), not just a household identifier (`serial`) but also a person identifier within each household (`pernum`). Hence, we can aggregate the data from the person level to the household level and subsequently to the block level. Using the variables above, we create 'geoblocks' consisting of household chunks with either 10, 25, or 50 households. Within every county of every state, to assign block identifiers, the data is sorted by the `reel`, `pageno`, and `lineno`. Subsequently, within counties in this order, every 10 / 25 / 50 households are given the same block identifier (hence going from 1 to the number of households divided by the block size).[5] The code of this exercise is available in:

1. `1_reel_page_line_household_blocking_part1_inspect.py`
2. `2_reel_page_line_household_blocking_part2_createblocks.py`
3. `3_reel_page_line_household_blocking_part3_createhouseholdblocks.py`

C Losses from Blocking and Unrepresentativity in Family Tree Genealogical Matches, 1900-1910

We use hand-made genealogy matches to measure the performance of our matches versus others in the literature. A prominent available dataset is Family Tree, which provides millions

[5] If within a county there are 235 households, for instance, and thus the last block is smaller, e.g., with only 5 households in the case of blocking by 10 households, then the last block is added to the block before that, thus the last block ID in the county then contains households 221 until 235.

of hand-made links between the 1900 and 1910 waves of the U.S. census. In their raw form, however, these matches are not necessarily representative of the general population. Figures 12 - 15 show that Family Tree genealogy matches tend to be skewed against people who move from their birth state, foreign-born residents of the US, non-white racial minorities, and certain age groups. These imbalances could skew, for example, the apparent share of people who move counties or switch occupations.

To address these imbalances we thus draw a more representative subsample of the Family Tree. We first take the full set of potentially-matchable individuals in the 1910 census. We consider someone to be potentially matchable if they were born no later than the year of the prior census wave (i.e. 1900) and, if they immigrated to the US, did so by the year of the prior census wave. From this set of people, we compute the share of individuals in each demographic cell among the following combination of variables:

- Gender, male vs female
- Race, white vs non-white
- State of birth if born in the US, or being foreign-born if not (without taking the exact country of origin)
- State of residence in 1910
- Age according to five-year age buckets, capped at 80

We tabulate the number of people in each such cell in the Family Tree genealogy matches, and assign an individual's probability of being drawn as the share of people in their census cell divided by the number of people in their cell in the Family Tree data. We then randomly draw a subsample of 1 million individuals. This significantly improves the representativity of the genealogy matches with regard to a handful of key demographic variables.

Within this more-representative subsample of the population, we calculate that 8% of men fail to meet one or more of the four strict blocking criteria mentioned in the main text (matching on birthplace, birth years at most 3 years apart, matching on first initials, and matching on last initials). Strict blocking thus immediately discards a non-trivial share of potential matches, whereas our algorithm aims to apply more flexible blocking criteria so as to incur fewer potential losses.

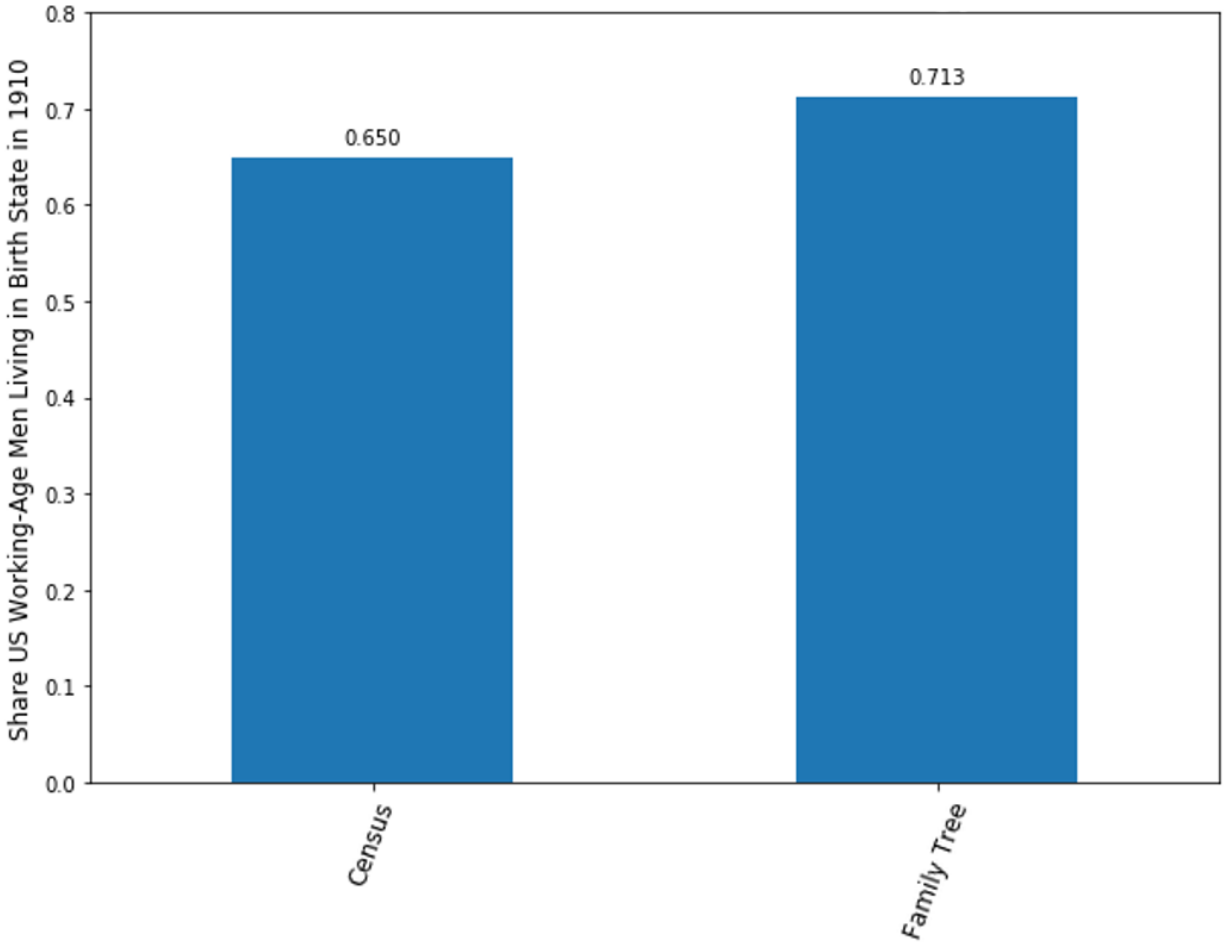


Figure 12: Anti-Mover Unrepresentativity in Family Tree Genealogy

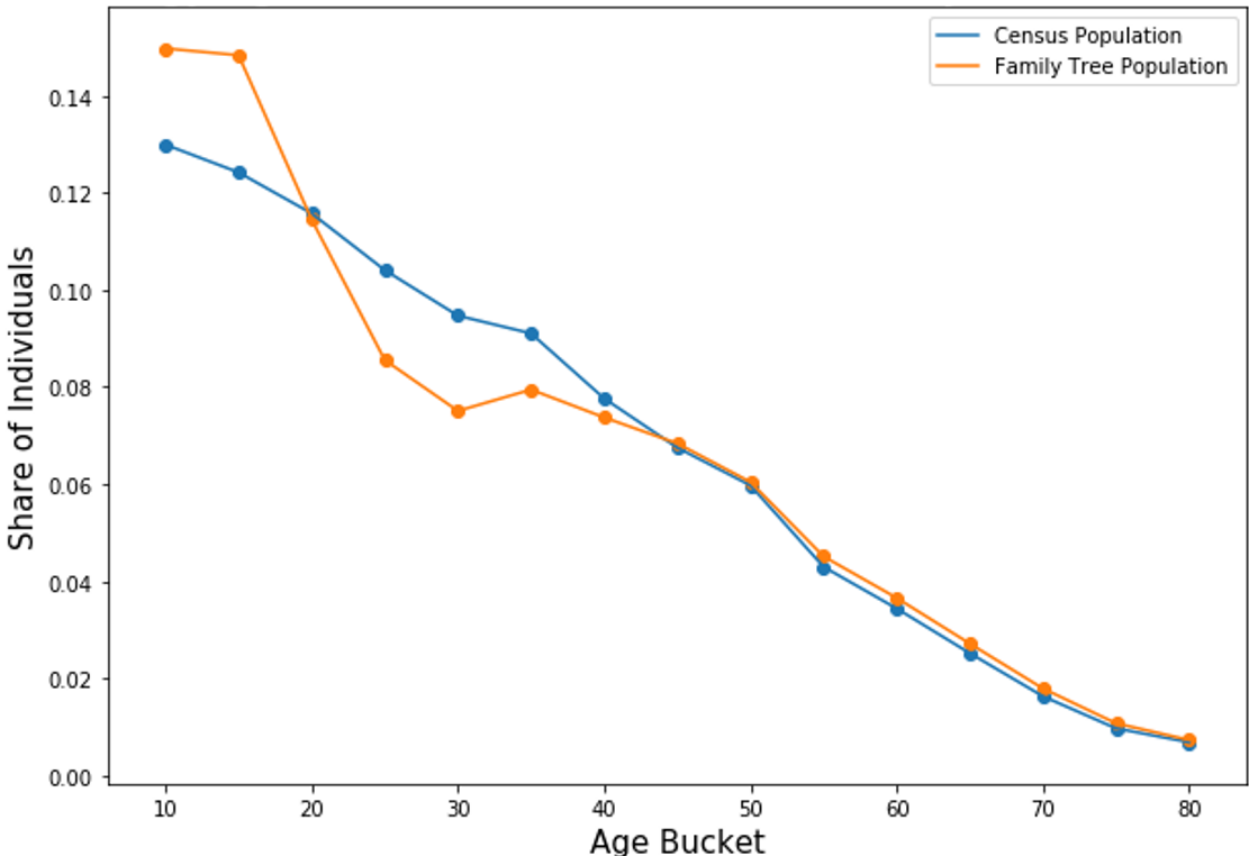


Figure 13: Age Unrepresentativity in Family Tree Genealogy

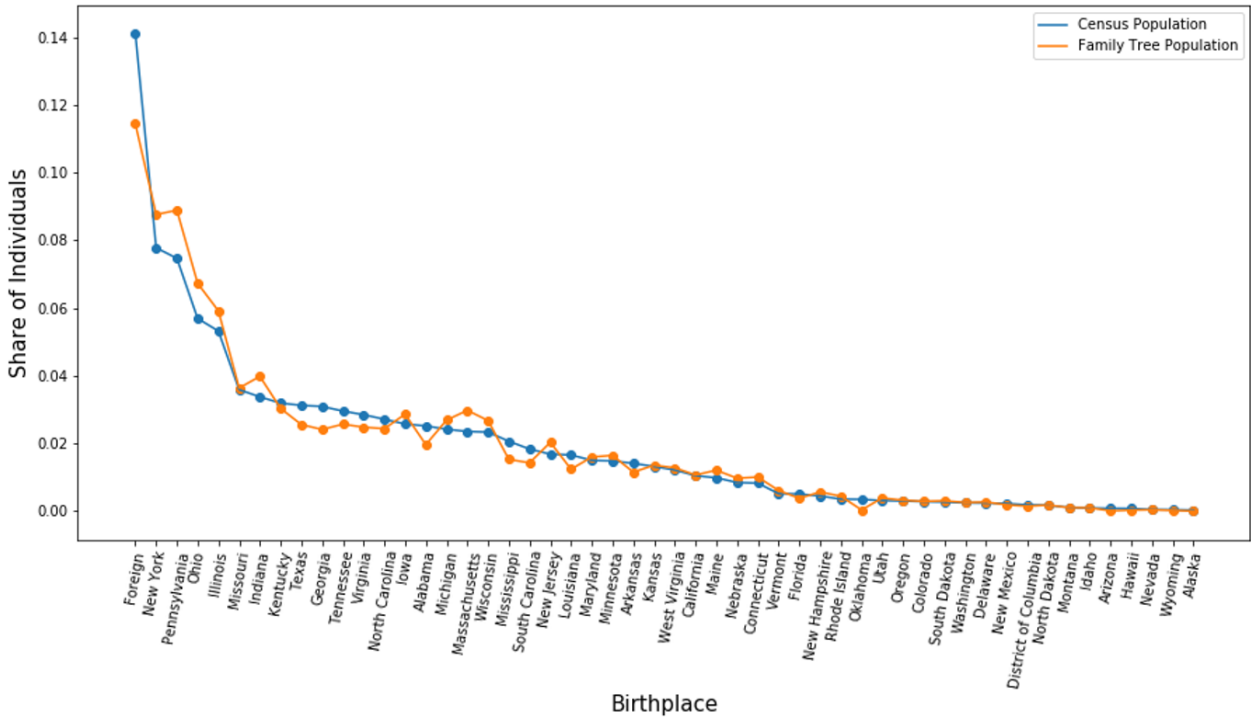


Figure 14: Birthplace Unrepresentativity in Family Tree Genealogy

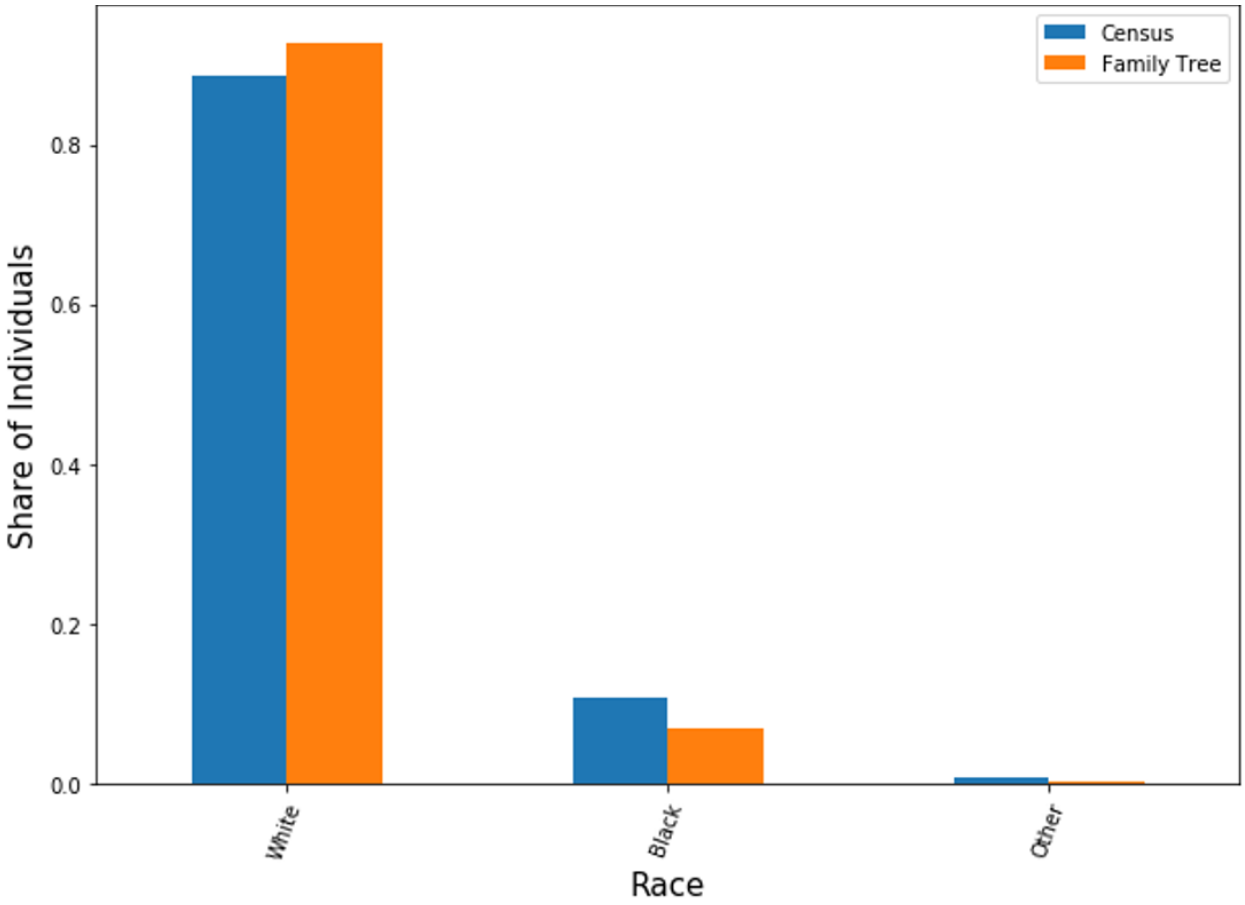


Figure 15: Racial Unrepresentativity in Family Tree Genealogy

D Household Matching for Ground Truth Creation

It has been noted previously in the literature (see e.g. Helgertz et al. (2022)) that leveraging family information to create individual matches could skew matches against people who change their household of residence, for example if they move or get married. We argue, however, that these family-based matches ought not to be skewed in terms of variables that remain constant over a person's entire life, namely their birthplace, year of birth, and name. Thus family-level information can be used to create a ground truth dataset for further training, as long as that ground truth dataset is only leveraged for these specific variables.

To assemble these ground truth matches, we filter for pairs of households across census waves that 1) have at least four individuals from Wave 1 with a match candidate in Wave 2, and vice versa and 2) have no such relationship with any other households. Figure 16 demonstrates this logic. Household A in Year 1 and Household D in Year 2 share five linked match candidates, and all their other possible linked households share fewer than four individuals. Requiring several match candidates in common reduces the probability of spuriously-matched households, and the uniqueness criterion ensures that the household-to-household match is likelier than any other alternative candidates.

When these two criteria are used together, requiring several match candidates in common

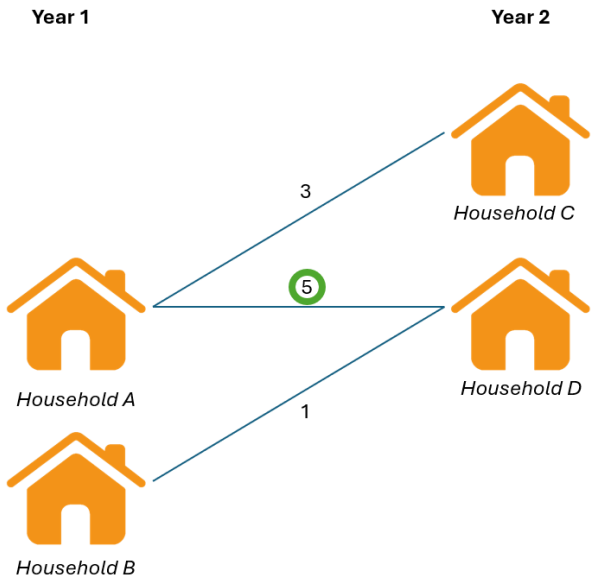


Figure 16: Household Matching Logic

also ensures that the matched households do not solely contain highly unique individuals. In our experiments, requiring just two individual matches in common (and requiring that no other candidate households reach this number of matches) results in individuals with extremely unique names and a far lower number of individual match candidates than the rest of the match candidate population. At four matches in common, however, individuals appear far more normal and have a number of match candidates close to the general population.

We operationalize this household matching procedure through sparse matrix operations. Let $\mathbf{M}_{12}$ be a sparse matrix of individuals in the first and second match waves, where entries are 1 if a pair of individuals form a match candidate; let $\mathbf{F}_1$ be a matrix of individual and family IDs in the first match wave, where entries are 1 if an individual is in a particular family; and let $\mathbf{F}_2$ be the equivalent individual-family matrix for the second match wave. Then we can execute the following algorithm:

1. Identify the number of individuals in each Wave 1 household who have at least one match candidate in a given Wave 2 household: $\mathbf{N}_{f1,f2} = \mathbf{F}_1 \cdot sign(\mathbf{M}_{12} \cdot \mathbf{F}_2^T)$, where $sign()$ converts all values above 0 to 1.

2. For a number of times one below the number of desired household-to-households matches (i.e. 3 in this case because we want 4 matches), compute $\mathbf{N}_{f1,f2} = \mathbf{N}_{f1,f2} - sign(\mathbf{N}_{f1,f2})$. This converts $\mathbf{N}_{f1,f2}$ to a binary-valued matrix where entries are 1 if there are at least 4 individuals from a Wave 1 household that each have at least one match in a Wave 2 household.

3. Repeat steps 1-2 going from Wave 2 to Wave 1 households rather than the reverse, yielding $\mathbf{N}_{f2,f1}$

4. Identify a pair of households as having at least 4 individuals matching between them if both $\mathbf{N}_{f1,f2}$ and $\mathbf{N}_{f2,f1}$ are satisfied. This ensures that there are in fact four distinct

people in each wave being matched, rather than, for example, four individuals from the Wave 1 household who all share the same match candidate in the household in Wave 2. Calculate $\mathbf{H}_{1,2} = \mathbf{N}_{f1,f2} + \mathbf{N}_{f2,f1}^{T}$ then $\mathbf{H}_{1,2} = \mathbf{H}_{1,2} - sign(\mathbf{H}_{1,2})$

5. Filter for households pairs where both households only have one matched household that satisfies the above conditions. We multiply $\mathbf{H}_{1,2}$ entries by their row and column sums, and filter for entries that equal 1.

6. Extract the resultant set of filtered matched households.

E Stage 2 Machine Learning Algorithm with Demographic Information

We load in ground truth and false matches. We take a sample of 100,000 ground truth matches, load in 300,000 ground false matches for individuals in the ground truth sample, and then eliminate 10% of the true matches (while retaining those individuals' false matches) so that the algorithm is forced to recognize the possibility of non-matching. We draw these proportions and sizes of training matches so that there is a reasonably large sample to train on, but not one so overwhelming as to incur excessive training times; and so that false matches outweigh true matches by a significant ratio, so that the algorithm learns to be somewhat discerning in identifying true matches.

We take and/or create the following variables for each pair of candidates:

- Jaro-Winkler Score for similarity of first names
- Jaro-Winkler Score for similarity of last names
- A binary indicator of whether first initials match
- A binary indicates of whether middle initials match
- A binary indicator whether first and middle initials in the first wave are flipped in the second wave (for example, A B in the first wave then B A in the second wave)
- A binary indicator of whether the first name is only listed as a single initial, either in the first or second wave
- A binary indicator of whether birthplaces match
- The difference in birth years between the candidates
- Age of the candidate in the first wave
- Age of the candidate in the second wave
- Gender of the candidate

We then train an xgboost machine learning algorithm that takes in data with these variables, and outputs a binary logistic prediction of whether the match is true or not with probability ranging from 0 to 1. We manually tune hyperparameters with a basic grid search, where we randomly split the data half and half into a training and out of sample dataset and optimize over the out-of-sample Matthews Correlation Coefficient. We also include a SMOTE procedure in our training algorithm to improve its performance on different values and combinations of variables.

F Additional Evaluations of Match Representativity of Static Demographic Variables

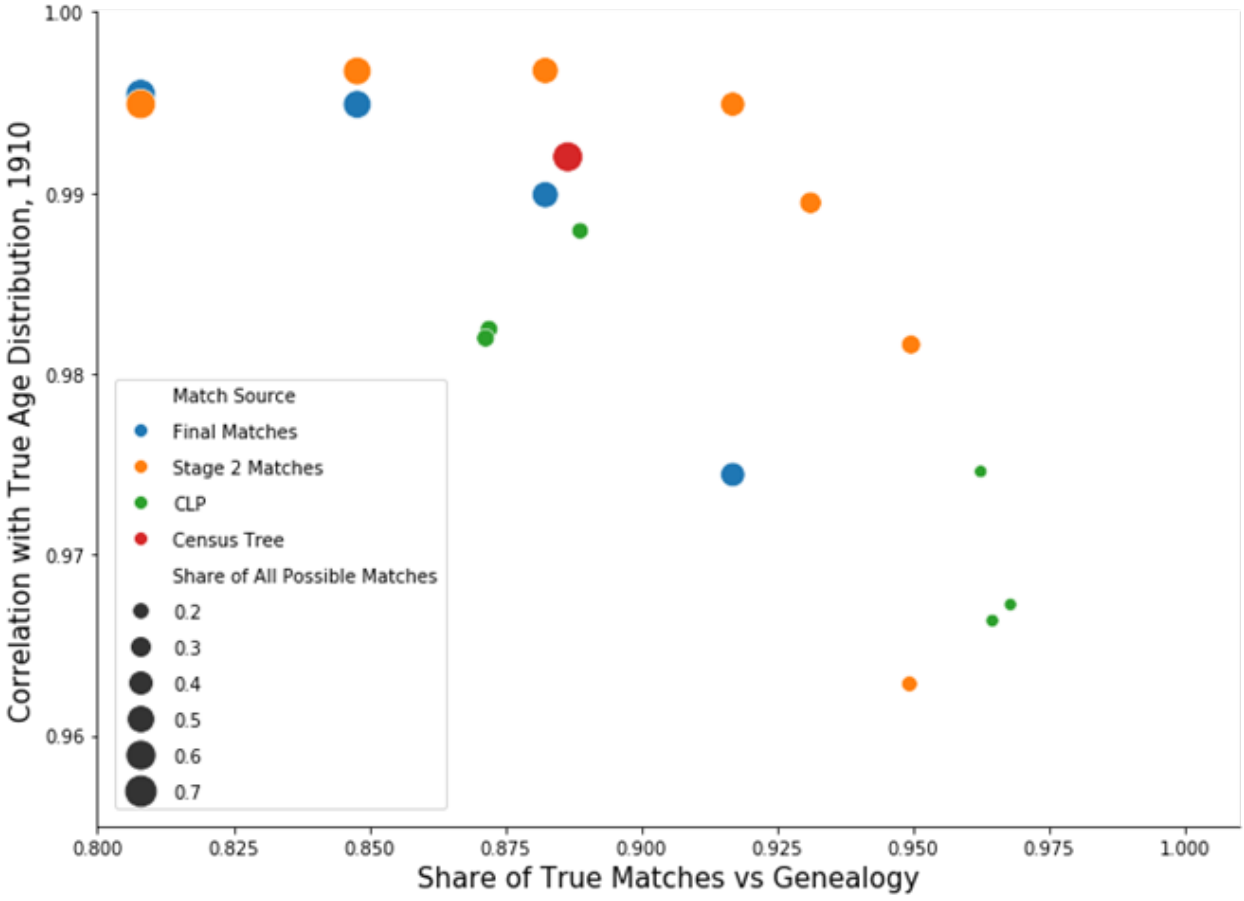


Figure 17: Match Comparison: Age Representativity for Men

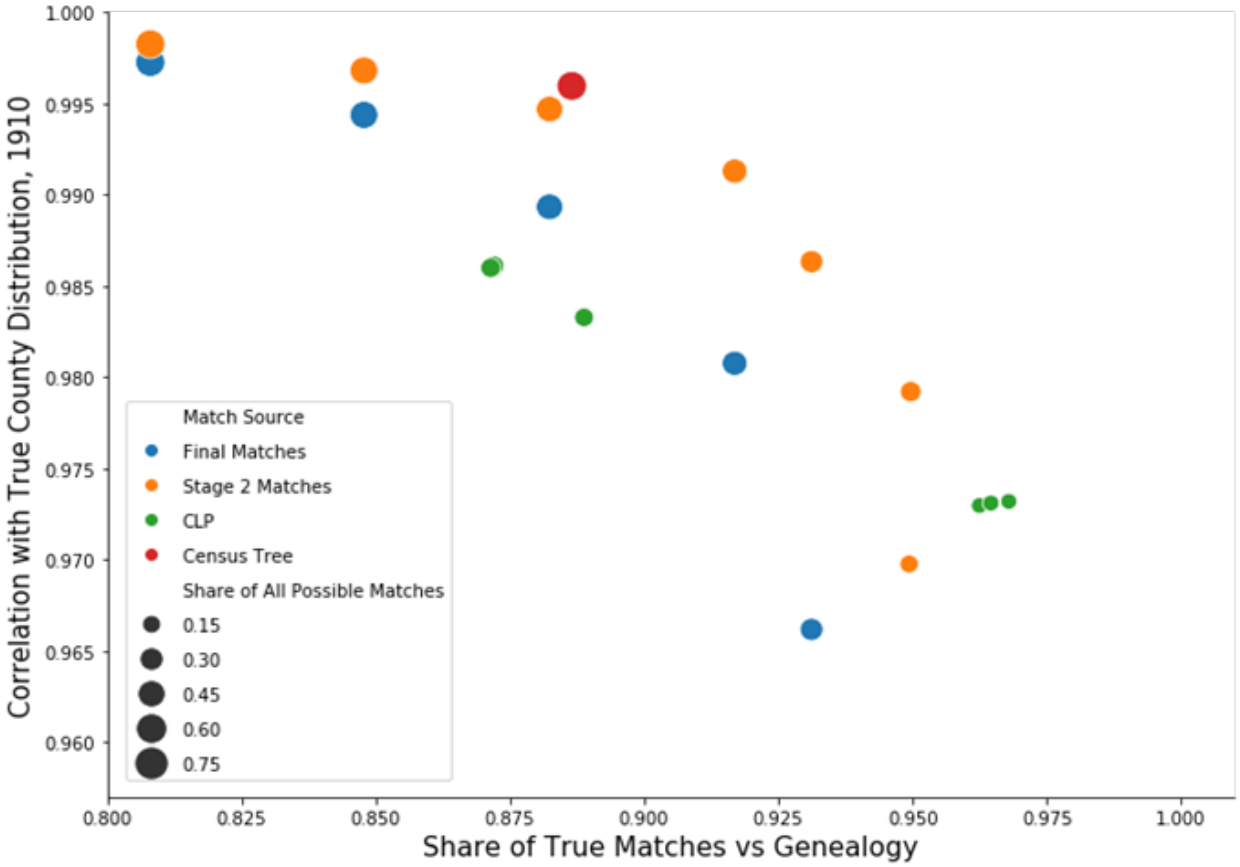


Figure 18: Match Comparison: County Representativity for Men

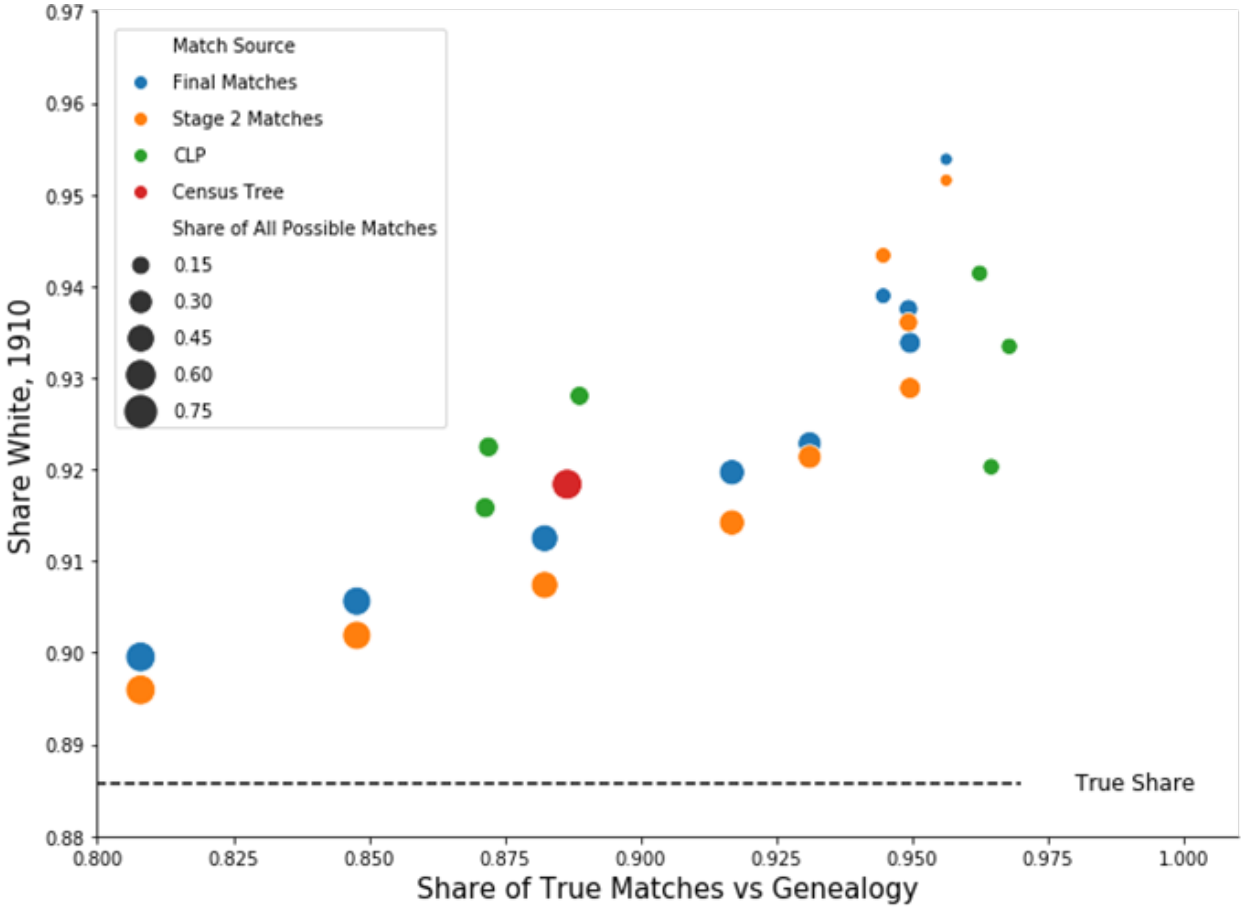


Figure 19: Match Comparison: Race Representativity for Men

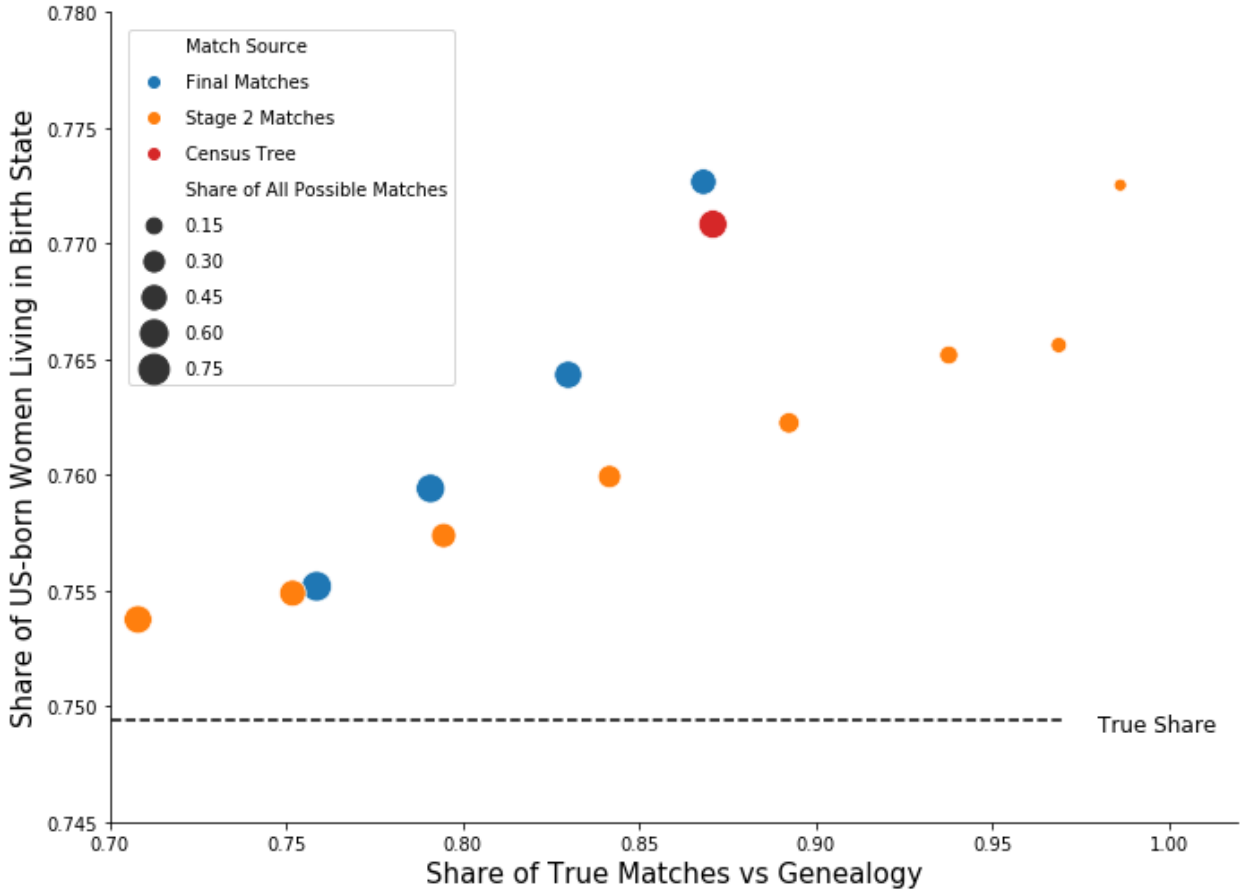


Figure 20: Match Comparison: Representativity of Living in Birth State for US-Born Women