**How to implement the ABE matching method**

The code used to run each of the following sections is contained in *ABE\_matching\_algorithm.do*. The following document explains each section of the code in more detail.

A note on large data sets:If you are using a large data set (e.g. the full-count census), prior to beginning the matching process it is best to save the data by place of birth. For example, if you are matching US born men in the 1900 census to the 1910 census, you should have 100 separate data files (50 states in each year). The following linking steps will then be completed for each place of birth separately (e.g. link Alabama in 1900 to Alabama in 1910, then Arizona in 1900 to Arizona in 1910, ect…)

A note on external packages: to use these codes you will need to download the following commands from ssc: “nysiis”, “jarowinkler”, “parallel”. Running the following lines in Stata would do the trick: “ssc install nysiis”.

A note on linking methods: this document describes how to implement the ABE method using our codes. For a more detailed presentation of this and other algorithms, check: **Abramitzky, Ran, Leah Boustan, Katherine Eriksson, James Feigenbaum, & Santiago Perez (2019). “Automated Linking of Historical Data”.** (<https://ranabr.people.stanford.edu/sites/g/files/sbiybj5391/f/linking_may2019.pdf>).

**Section 1: Standardize variables and clean raw names**

* 1. To make the matching process easier, first standardize the variable names to match those used in this code.
  2. The stata command *abeclean* cleans raw names and generates NYSIIS standardized names.

**Section 2: ABE method using exact names or nysiis standardized names**

2.1 The ABE algorithm with exact names and nysiis standardized names follows the same process. At this step of the matching process there is the option to either match on exact name or match on NYSIIS standardized name. There is also the option of including additional time-invariant matching variables such as middle names, race, or place of birth. The best choice of matching variables will be context-specific. All matching variables, other than age, should be included in the global “match\_vars”. The global “timediff” is the number of years between the reported age in each data set. If you are using year of birth rather than reported age, simply set “timediff” equal to 0.

2.2  *Standard version:* In the standard version individuals are required to be unique by name (either exact name or NYSIIS standardized name) within each year of birth. The stata command abematch.ado produces a file containing all successful matches from the standard ABE method. Any variables included in the “keep\_A” and “keep\_B” will be included in the output file, with the suffix “\_A” or “\_B”.

2.3 *5-year band:* In this version if there is more than one potential match for an observation within ± 2 ears of age we drop this record from our analysis. Additionally, we require that individuals are unique by name (either exact or NYSIIS) within a 5-year age band in terms of year of birth. The variable unique\_file2 marks matches that are not unique within +-2 years in their own data set. The variable unique\_match2 marks matched pairs that have another potential match within +-2 years of birth.

**Section 3: ABE using Jaro-Winkler adjustment**

This version of the ABE algorithm is analogous to the process used in Section II, but rather than finding pairs that match on exact or standardized name, this version uses the Jaro-Winkler string distances between each potential pair of names to determine which records to match.

3.1 This approach requires finding a Jaro-Winkler string distances between the names of every potential match. In small data sets it may be feasible to find the distance between all names, but for larger data sets we need to put some limits on what pairs of names we find the string distance for. One way we do this is block the data into smaller chucks, and only find the JW string distance between names with the same block. One common method is to block on the first letter of first and last names, since there is reason to believe that errors in name spelling and transcription are less likely to occur with the first letter. The data could be further blocked on place of birth, race, or other time-invariant characteristics. We typically find the string distances for all names within ± 5 years of reported age (age\_band = 5). This allows us to apply the 5-year uniqueness band requirement (x = 2) in step 3.3[[1]](#footnote-1).

3.2 Use the pairing command to find the JW string distance between the names of all potential matches. Note that the global “timediff” specifies the number of years between the reported age given in data set A and data set B, and will need to be updated to fit your context. The input file “gen\_initals.do” splits the name variables to find initials. The file “filtering.do” keeps only pairs within the specified age band and with JW string distances of <= 0.1 in first (middle) and last name.

3.3 Apply ABE-JW decision rules using *JW\_decision\_rules.do.* Once we have found a list of all potential matches (pairs within the specified age bandand who have a JW distance in first, (middle) and last names of ≤ 0.1) there are three possible scenarios:

1. We find no potential match for a given observation in data set A. In this case this individual is dropped from our analysis.
2. We find only one potential match for an individual in data set A, and the corresponding individual in data set B also has no other potential matches. In this case we consider the pair to be a successful match.
3. We find more than one “name match” for an observation in the source data. In this case, we use the following decision rule:
4. We compute the absolute value of the age difference with respect to the observation with the closest age (let’s call this difference d1).
5. We compute the absolute value of the age difference with respect to the observation with the second closest age (let’s call this difference d2).
6. We keep only individuals such that d2-d1>x, such that larger values of x represent more conservative linking choices. In other words, we choose the closest observation with respect to age, but only if the second closest observation is “far enough”, where the value of x determines how far enough we want the second closest observation to be.  The choice of x is up to the researcher, but we recommend testing robustness to more and less conservative decision rules whenever possible.

Any value of x can be used, however we typically show results using both x = 0 and x = 2. When x = 0 the closest match in terms of age difference is considered a successful match, so long as the match is unique. When x = 2 we drop any observations with more than one potential match within +- 2 years of reported age. At this step we also specify the number of years reported age can differ between matches (max\_age\_diff). We typically report results allowing age differences ≤ 2 years (max\_age\_diff = 2), but we also recommend testing results when requiring pairs to match on exact age (max\_age\_diff = 0).

3.4 Apply within data set uniqueness requirements. Since JW string distances are not transitive, in some cases records will successfully match by JW string distance, even though there is another name within his/her own data set within 0.1 JW string distance in name. To drop these cases, we need to find string distances between the names of each person within his/her own data set.

* + 1. The code to find these “within dataset” string distances and identify the uniqueness of each match is contained in JW\_within\_dataset\_uniqueness.do.
* First we find the string distances between all name from data set A that we were able to match (i.e. all people in the least restrictive matched sample, x = 0) and the names of all other people in data set A within ± 2 years of age. Repeat the same process for data set B.
* Next we identify which matched observations are non-unique by JW string distance within their own data set. We create an indicator for those who are non-unique by exact age, and those who are non-unique with in ± 2 years of age.

3.4.2. Drop all matched pairs that are non-unique by JW string distance within specified age bands. We typically require that names are unique by exact age (0 yr band) when x = 0, and are unique within +-2 years when using x = 2.

A note on output: The JW matching code will produce a data set giving a list of id variable(s) from each data file for every successfully matched pair. These id variables can then be merged back into the original data to attach any variables necessary for analysis.

Last updated: 2/14/20

Please contact ranabr@stanford.edu (Ran Abramitzky), lboustan@princeton.edu (Leah Boustan), and/or kaeriksson@ucdavis.edu (Katherine Eriksson) with any questions or feedback about this code.

1. To make the ABE-JW approach with x = 2 equivalent to the 5-year uniqueness band used in the ABE method with NYSIIS or exact names, search for potential matches only within ± 2 years of reported age. This will give a slightly less conservative matched sample. [↑](#footnote-ref-1)