SAS® GLOBAL FORUM 2021

Paper 1048

It's In My DNA: A SAS® Program to Assist with Finding Unknown Relatives

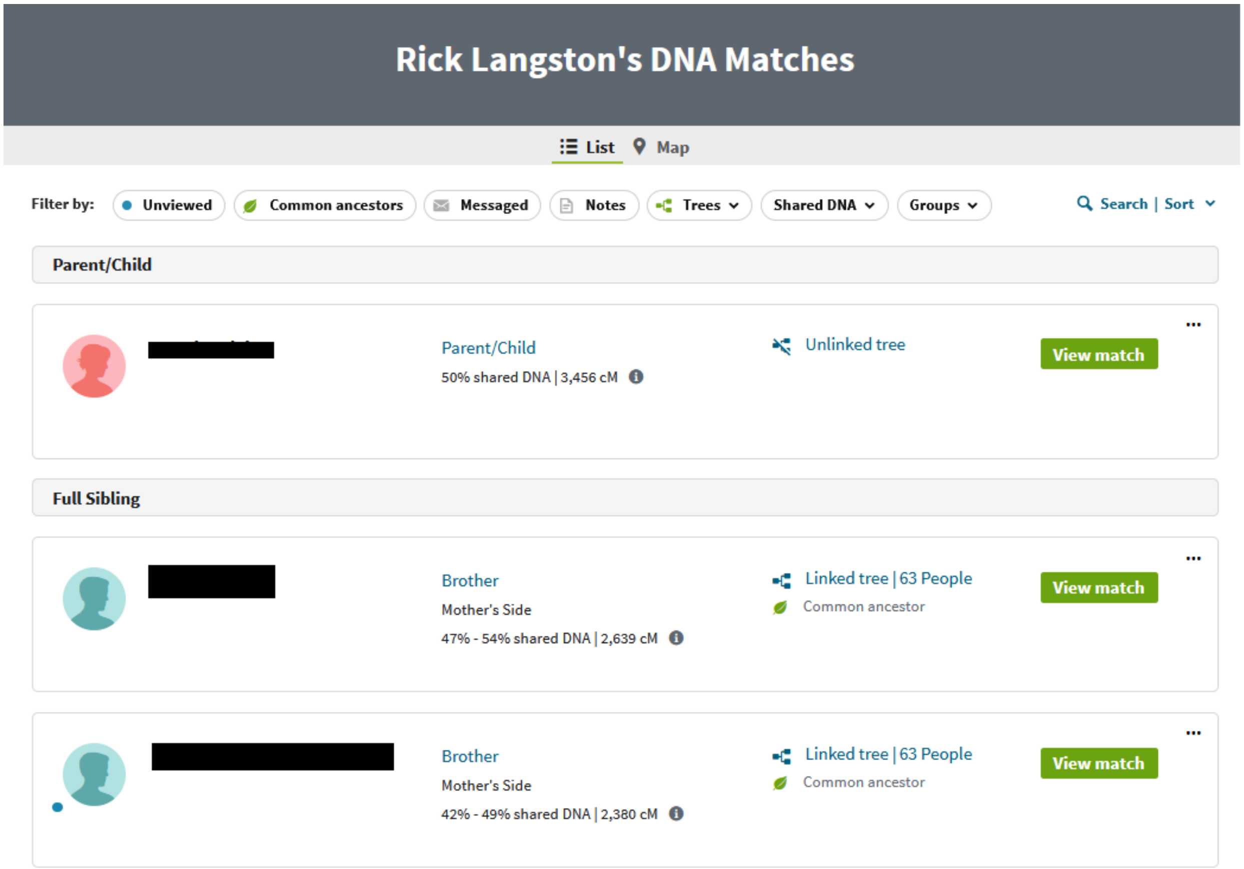
Richard D. Langston, Retired

ABSTRACT

This paper describes a SAS program I created to process DNA matches from Ancestry.com and which assists the user in creating descendancy listings to help determine previously unknown biological parents. The program reads HTML data to extract match results, and provides a mechanism to cross-reference these results with relationships constructed from family trees, obituaries, and other historical records.

TAKING A DNA TEST VIA ANCESTRY.COM

Many readers, especially those in the US, may have already taken a DNA test. There are many providers of these tests, with the most popular being Ancestry.com. You can purchase a DNA kit from them, simply spit in a tube and mix it with a provided chemical, pop it in an envelope and mail it off. I did that myself, and about six weeks later, I could see the DNA match profiles as per Figure 1.



Rick Langston's DNA Matches

List | Map

Filter by: ● Unviewed | Common ancestors | Messaged | Notes | Trees | Shared DNA | Groups | Search | Sort

Parent/Child

[redacted]
Parent/Child
50% shared DNA | 3,456 cM
Unlinked tree
View match

Full Sibling

[redacted]
Brother
Mother's Side
47% - 54% shared DNA | 2,639 cM
Linked tree | 63 People
Common ancestor
View match

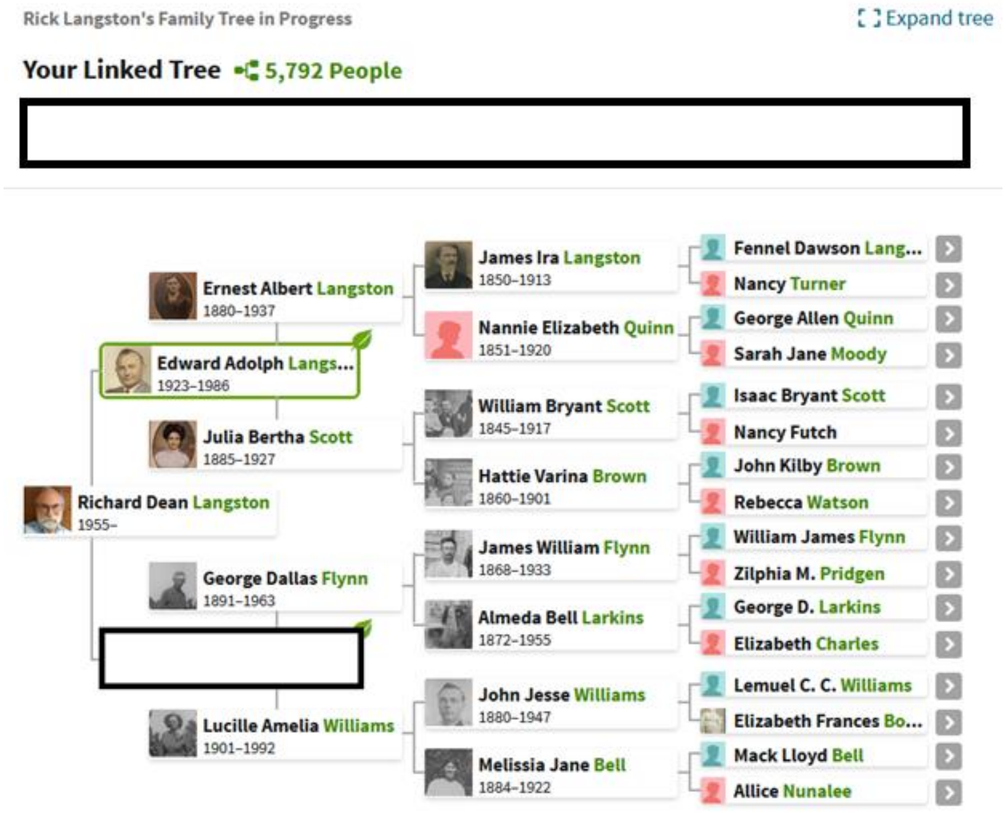
[redacted]
Brother
Mother's Side
42% - 49% shared DNA | 2,380 cM
Linked tree | 63 People
Common ancestor
View match

Figure 1. First Page of DNA Match Profiles

(Note that all names except mine have been redacted in this and subsequent figures, to protect the privacy of individuals.)

This page shows that I have a 3456 centimorgan match with my mother (which is in the expected range) and I have matches of 2639 and 2380 with my brothers (also expected). You can use the DNA Painter website (link in the References section) to get a better understanding of what these centimorgan counts mean and what relationships are most likely based on these counts.

You can also see links to trees, which are very important when trying to understand relationships for people you might not otherwise know. Figure 2 shows my tree, which has been linked to my DNA profile and to that of my brothers and mother.



Rick Langston's Family Tree in Progress

Your Linked Tree 5,792 People

[redacted]

Richard Dean Langston
1955–

Edward Adolph Langs...
1923–1986

[redacted]

Ernest Albert Langston
1880–1937

Julia Bertha Scott
1885–1927

George Dallas Flynn
1891–1963

Lucille Amelia Williams
1901–1992

James Ira Langston
1850–1913

Nannie Elizabeth Quinn
1851–1920

William Bryant Scott
1845–1917

Hattie Varina Brown
1860–1901

James William Flynn
1868–1933

Almeda Bell Larkins
1872–1955

John Jesse Williams
1880–1947

Melissia Jane Bell
1884–1922

Fennel Dawson Lang...
Nancy Turner
George Allen Quinn
Sarah Jane Moody
Isaac Bryant Scott
Nancy Futch
John Kilby Brown
Rebecca Watson
William James Flynn
Zilphia M. Pridgen
George D. Larkins
Elizabeth Charles
Lemuel C. C. Williams
Elizabeth Frances Bo...
Mack Lloyd Bell
Allice Nunalee

Figure 2. My Associated Family Tree For My DNA Match Profile

I'M A SEARCH ANGEL USING THE SAS SYSTEM

Now that you can see what is available to you once you've taken the DNA test, I can explain what I've been doing to assist people who take the test.

Many people take the DNA test in order to determine unknown relatives. They may be adopted, or they may have learned that their father is not who they thought he was. But after taking the DNA test, they may be overwhelmed with the data and lack the skill on how to interpret the results.

Researchers who help these people are often called "search angels" and that's a role I've taken on. And in the process of my search angel role, I found it useful to create a SAS program – originally called read_matches.sas – in order to help me manage the data and research. That program has been through several iterations, and is now called read_matches4.sas.

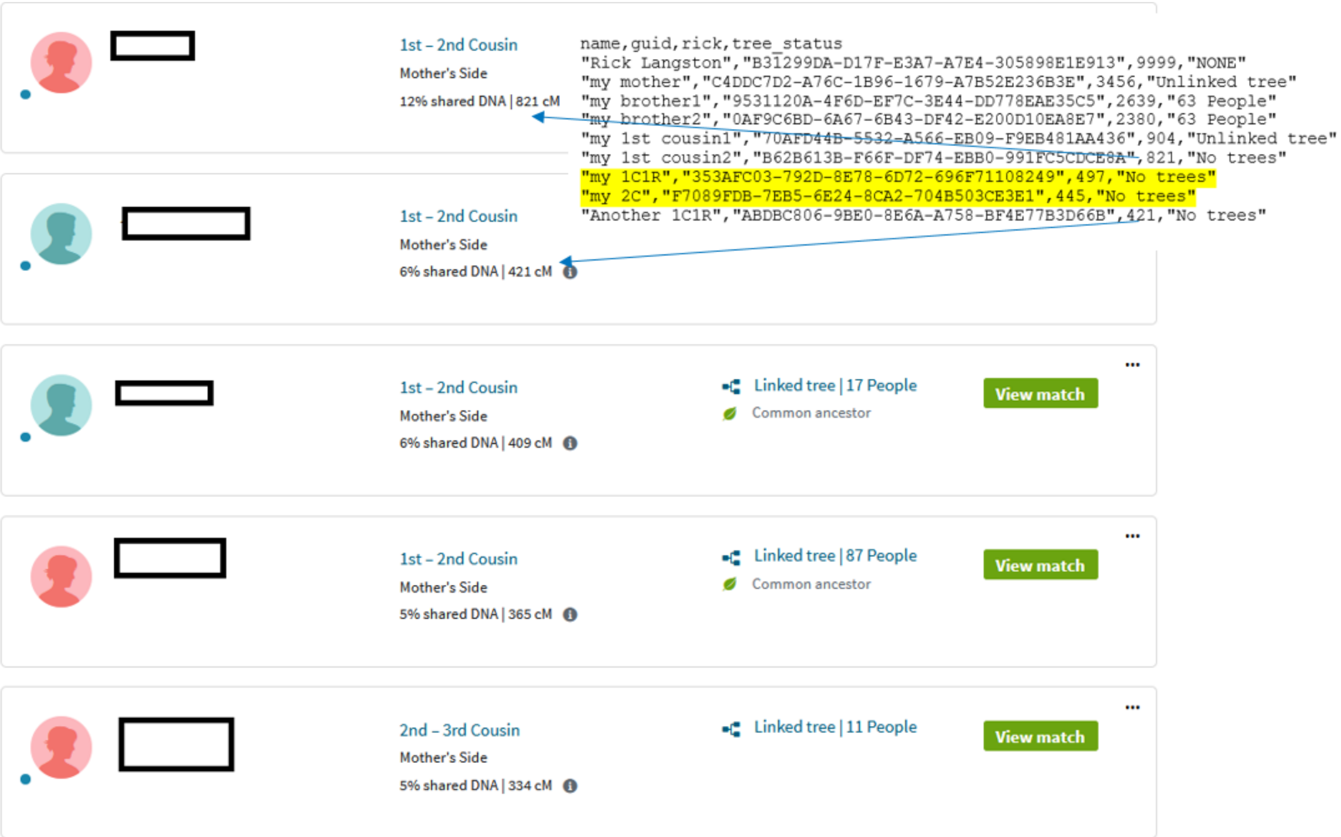
The read_matches4 program reads from the pages of matches (such as figure 1 above) and extracts pertinent data. Note that the HTML stream for the pages can be derived via the Firefox browser. I used the Ctrl-A key to "select all" then right-clicked on one of the highlighted fields. A drop-down menu appears which includes the selection "View Selection Source". That then displays the entire HTML stream, which can then be pasted into the Windows clipboard and then saved to a file.

That stream contains the values displayed on the page (name, centimorgan count, tree status) along with a unique ID which is not displayed – but contained within the HTML –

which is very important to the operation of the program. The unique ID, also called a UUID or GUID, is associated with the individual DNA profile, and is the proper key to use when managing the data because it is unique to each profile. There is no guarantee on the uniqueness of names (e.g. multiple John Smiths can appear) but the GUID will be unique.

The program generates a comma-separated file (CSV) containing the pertinent fields, which are: name, GUID, centimorgan count, and tree status. In Figure 3, we can see another page of matches, associating various lines in the CSV with the corresponding page entries. (Note that the GUIDs in this figure are contrived and are not the actual GUIDs for these entries).

Note also that the column name for centimorgan count is the name associated with the CSV itself. In this example, the column name is rick, corresponding to my match count values. Each one of the CSVs will have a different name in that column, in order to allow for merging based on the unique GUID.



[redacted]
1st – 2nd Cousin
Mother's Side
12% shared DNA | 821 cM

```
name,guid,rick,tree_status
"Rick Langston","B31299DA-D17F-E3A7-A7E4-305898E1E913",9999,"NONE"
"my mother","C4DDC7D2-A76C-1B96-1679-A7B52E236B3E",3456,"Unlinked tree"
"my brother1","9531120A-4F6D-EF7C-3E44-DD778EAE35C5",2639,"63 People"
"my brother2","0AF9C6BD-6A67-6B43-DF42-E200D10EA8E7",2380,"63 People"
"my 1st cousin1","70AFD44B-5532-A566-EB09-F9EB481AA436",904,"Unlinked tree"
"my 1st cousin2","B62B613B-F66F-DF74-EBB0-991FC5CDCE8A",821,"No trees"
"my 1C1R","353AFC03-792D-8E78-6D72-696F71108249",497,"No trees"
"my 2C","F7089FDB-7EB5-6E24-8CA2-704B503CE3E1",445,"No trees"
"Another 1C1R","ABDBC806-9BE0-8E6A-A758-BF4E77B3D66B",421,"No trees"
```

[redacted]
1st – 2nd Cousin
Mother's Side
6% shared DNA | 421 cM

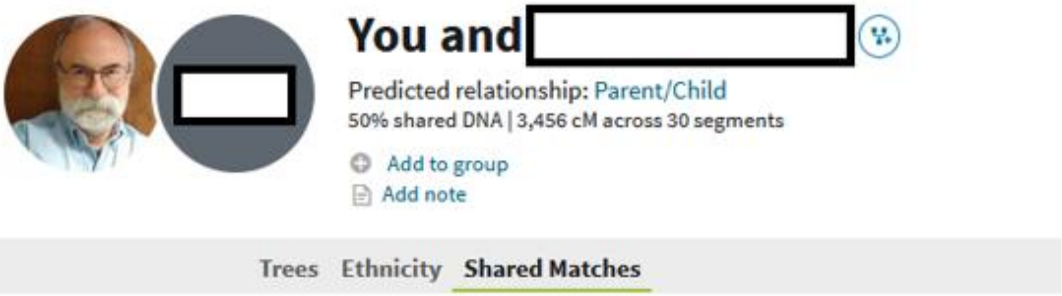
[redacted]
1st – 2nd Cousin
Mother's Side
6% shared DNA | 409 cM
Linked tree | 17 People
Common ancestor
View match

[redacted]
1st – 2nd Cousin
Mother's Side
5% shared DNA | 365 cM
Linked tree | 87 People
Common ancestor
View match

[redacted]
2nd – 3rd Cousin
Mother's Side
5% shared DNA | 334 cM
Linked tree | 11 People
View match

Figure 3. CSV Records And Associated DNA Profiles

An important feature of DNA matching on Ancestry.com is Shared Matches. With this feature, you can indicate any of your matches for whom you want to see all DNA profiles that you have in common. It will display a panel that looks like Figure 4 below. This begins the page that shows shared matches with my mother.



[redacted]

You and [redacted]

Predicted relationship: Parent/Child

50% shared DNA | 3,456 cM across 30 segments

Add to group

Add note

Figure 4. Shared Match Panel

In Figure 3 above, which shows shared matches between my mother and me, we can see arrows pointing from the CSV line for a first cousin with 821 cM matching, and we see the next match for another cousin with 421 cM. But there are two CSV records highlighted in yellow that do not appear in the list of DNA profiles. That is because those two profiles are not shared matches with my mother, meaning that instead I am related to them on my father's side.

And this is the important point to understand when you are trying to determine unknown relatives. You need to know who is related on your father's side and your mother's side. These will be distinct family lines that will be divided up to be researched separately. So the known shared matches between you and other DNA profiles are very important, and those should be placed in separate CSV files.

FIRST OUTPUT FROM THE PROGRAM

(Note that for subsequent discussion in this paper, contrived DNA profiles will be used.)

Once the program has created that first CSV – a portion of which is shown in figure 3 above – it reads that first CSV and produces an output that shows the initial matches. The output looks like Figure 5 below. We see a name column, a column for the primary person (in this case John) showing centimorgan match counts with each other person, the tree status, and the GUID. But also seen is a CRC. This is a Cyclical Redundancy Check hash, which gives us a reduced-size version of the GUID. We want a smaller version of the GUID for a less crowded data stream that we will see later in this paper.

```
x     name            john   tree_status          crc       guid
    1 John Smith    (SELF)   100 People           FA10E4D1  B31299DA-17FE-A7A7-4305-98E1E913C4DD
    2 Mary Jones      1000   50 People            8487E349  C7D2A76C-B961-79A7-52E2-6B3E9531120A
    3 Bill Johnson     900   No Trees             457C2C02  4F6DEF7C-E44D-778E-E35C-0AF9C6BD6A67
    4 James Johnson    500   Locked               562EDDE4  6B43DF42-200D-0EA8-770A-D44B5532A566
    5 Ann Peterson     400   Unlinked Tree        7BFC95BF  EB09F9EB-81AA-36B6-B613-F66FDF74EBB0
    6 Randy Jones      300   20 People            BE1B9664  991FC5CD-E8A3-3AFC-3792-8E786D72696F
    7 Angela Smith     300   2 People             6784051D  71108249-7089-DB7E-56E2-8CA2704B503C
    8 Kim Madison      250   No Trees             73723DB3  E3E1ABDB-8069-E08E-AA75-BF4E77B3D66B
    9 Peter Wilson     200   123 People           196D8AF7  263E6EF0-8D19-1CB6-4352-5C9D0CE0421C
```

Figure 5. Initial Output From The Program

We see that the person next in the list of DNA matches is Mary Jones. So we will want to view the page of shared matches between John and Mary. We capture that data like before and create another CSV. With the next run of the program, it will look for all available CSV files and will read all of them to produce updated output, including shared matches with

Mary. We will see output that looks similar to Figure 6. (Note that the GUIDs are not being shown in subsequent figures to allow for less clutter in the figures.)

```
x     name           john   mary   tree_status           crc
    1 John Smith   (SELF)   1000   100 People            FA10E4D1
    2 Mary Jones     1000 (SELF)   50 People             8487E349
    3 Bill Johnson    900      _   No Trees              457C2C02
    4 James Johnson   500      _   Locked                562EDDE4
    5 Ann Peterson    400      Y   Unlinked Tree         7BFC95BF
    6 Randy Jones     300      _   20 People             BE1B9664
    7 Angela Smith    300      Y   2 People              6784051D
    8 Kim Madison     250      _   No Trees              73723DB3
    9 Peter Wilson    200      Y   123 People            196D8AF7
```

Figure 6. Next Output With Shared Matches

With this output, we can see that Mary has a shared match with Ann, Angela, and Peter. Note that we don't know the actual shared match count, since we don't have access to that data. We simply know that they are a shared match, so that is why we see Y for those cells. For the others in the column for Mary, we see an underscore. This means that Mary is not a shared match, and we should consider viewing a shared match page between those lines and John Smith. This may possibly show additional family lines. We see Bill Johnson is the first person with an underscore, so we can build a shared match CSV for him and run again. And that output appears in Figure 7. We will now see some columns with the standard dot missing value, meaning that the person is a shared match with some other column. And we can begin to see a pattern of who is related to whom. This process of repeated matching is known as the Leeds Method, created by Dana Leeds.

```
x     name           john   bill   mary   tree_status           crc
    1 John Smith   (SELF)     900   1000   100 People            FA10E4D1
    2 Mary Jones     1000      . (SELF)   50 People             8487E349
    3 Bill Johnson    900 (SELF)      .   No Trees              457C2C02
    4 James Johnson   500      Y      .   Locked                562EDDE4
    5 Ann Peterson    400      .      Y   Unlinked Tree         7BFC95BF
    6 Randy Jones     300      _      _   20 People             BE1B9664
    7 Angela Smith    300      Y      Y   2 People              6784051D
    8 Kim Madison     250      Y      .   No Trees              73723DB3
    9 Peter Wilson    200      .      Y   123 People            196D8AF7
```

Figure 7. Next Output With Still More Shared Matches

DESCENDANCY LISTING CONCEPT

Once we have established some family line understanding via the Leeds method, and created meaningful output as seen in the previous section, then we begin the work involving genealogical research.

The trees that are associated with the various DNA profiles are examined to look for common ancestors. Additional trees that are available at Ancestry.com, plus other genealogical records, such as obituaries, can also be used in order to get a better understanding of how the various matches are connected together.

The general consensus among DNA researchers seems to be that one starts building trees in order to help find unknown relatives. However, my experience has shown that it is better to go in the opposite direction and build a descendancy listing. This means that you start with a common ancestor and show all known children for each generation. Figure 8 shows an example descendancy listing. Two separate ancestors, Josiah Smith and George Avery begin the descendancies. Each later generation is indented two spaces. We can see that Randolph, Quincy, and William Smith are all grandchildren of Josiah. And we can see that one of George Avery's daughters, Lucille, is married to William Smith. So their children John and

Edward are connected to both descendancies and have both Josiah Smith and George Avery as ancestors.

```
Josiah Smith 1850-1920 + Eliza Fairfield 1852-1932
  Charles Smith 1883-1950 + Eleanor James 1887-1965
    Randolph Smith 1925-1990 + Elizabeth Nelson 1930-1998
      Angela Smith 1948-
  Raeford Smith 1885-1946 + Wilma Adamson 1890-1962
    Quincy Smith 1918-1995 + Mary Floyd 1923-2010
      Mary Smith 1955- + Phillip Jones
      Elizabeth Smith 1957- + Martin Walker
        Frederick Walker 1990-
    William Smith 1920-1990 + Lucille Avery 1922-2000
      John Smith 1950-
      Edward Smith 1952-

George Avery 1882-1940 + Ina McDonald 1886-1956
  Diane Avery 1919-1997 + George Johnson 1923-1998
    William "Bill" Johnson 1955- + Maria Simpson 1961-
      James Johnson 1992-
      Amelia Johnson 1996-
  Kenneth Avery 1920-1930
  Lucille Avery 1922-2000 + William Smith (see above)
```

Figure 8. A Descendancy Listing Example

At this point, we can connect the descendancy listings with the DNA match information. We do this by adding the CRCs for DNA matches with brackets into the proper lines of the listing. In figure 7 above, we see Angela, Mary, and John all in our cross-reference listing with their CRCs, and we can add those CRCs into the listing as seen in red in figure 9. Being able to insert smaller amounts of text (the CRC) instead of the 36-character GUID makes the descendancy listing more readable to human eyes.

```
Josiah Smith 1850-1920 + Eliza Fairfield 1852-1932
  Charles Smith 1883-1950 + Eleanor James 1887-1965
    Randolph Smith 1925-1990 + Elizabeth Nelson 1930-1998
      Angela Smith 1948- [6784051D]
  Raeford Smith 1885-1946 + Wilma Adamson 1890-1962
    Quincy Smith 1918-1995 + Mary Floyd 1923-2010
      Mary Smith 1955- [8487E349] + Phillip Jones
      Elizabeth Smith 1957- + Martin Walker
        Frederick Walker 1990-
    William Smith 1920-1990 + Lucille Avery 1922-2000
      John Smith 1950- [FA10E4D1]
      Edward Smith 1952-
```

Figure 9. Descendancy Listing With CRC Key Association

Once that match-up is in place, read_matches4 can be run again. Note that at this point its input includes not only the CSV files but also the descendancy listing seen in figure 9. The program will substitute match information into the output. An example of that appears in figure 10. We can see that Angela is a match with John at 300 cM and also a match with Bill and Mary. These was evident in the cross-reference in figure 7, but now we can apply our understanding of the meaning of centimorgan counts to see how they match with the actual known relationship of Angela, Mary, and John. We can see that Angela and Mary are second cousins. Mary is first cousin to John.

```
Josiah Smith 1850-1920 + Eliza Fairfield 1852-1932
  Charles Smith 1883-1950 + Eleanor James 1887-1965
    Randolph Smith 1925-1990 + Elizabeth Nelson 1930-1998
      Angela Smith 1948- [john:300,bill:Y,mary:Y]
  Raeford Smith 1885-1946 + Wilma Adamson 1890-1962
    Quincy Smith 1918-1995 + Mary Floyd 1923-2010
      Mary Smith 1955- [john:1000,bill:.,mary:(SELF)] + Phillip Jones
      Elizabeth Smith 1957- + Martin Walker
        Frederick Walker 1990-
```

```
William Smith 1920-1990 + Lucille Avery 1922-2000
  John Smith 1950- [john:(SELF),bill:900,mary:1000]
  Edward Smith 1952-
```

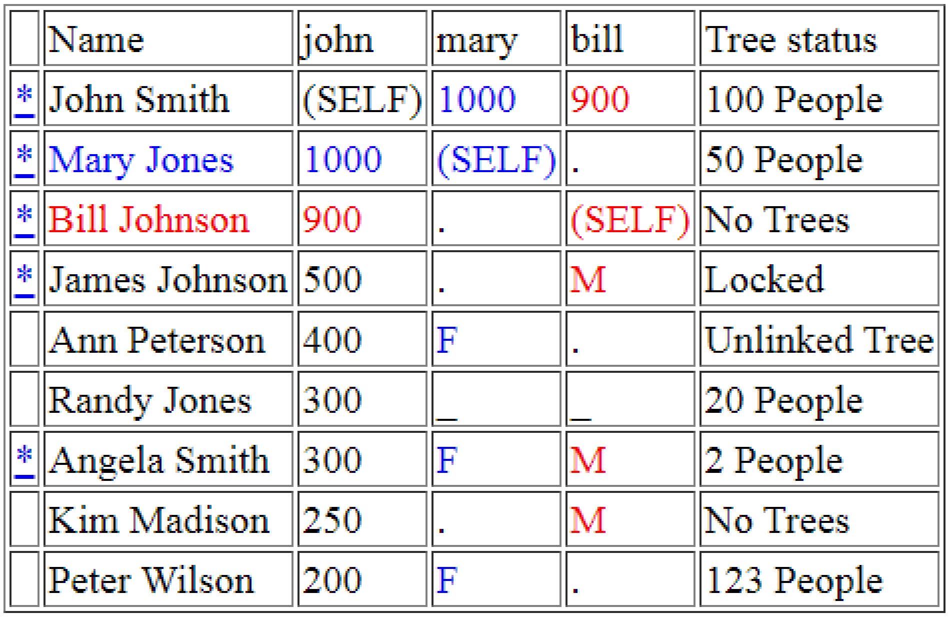
Figure 10. Output From The Program Substituting Match Information

If it had been the case that we did not know John's actual relationship, we could use the other known relationships as clues as to his placement in the listing.

As you might expect, this becomes an iterative process of continued genealogical research to expand out the descendancy listings to try and match up as many people as you can in order to determine unknown relatives.

HTML OUTPUT

Read_matches4 can produce textual output, seen in various figures above. But it can also produce HTML output, which may be more easily readable to many. And because the Leeds method also suggests the use of colors to help determine different family lines, the program can do that too. Figure 11 shows the browser rendering of an HTML table. The asterisks indicate that the entry appears in the descendancy listing, and in fact is a link to the proper place in the descendancy listing. The mother's side is shown in red and the father's side is shown in blue, and the letter Y is replaced with M (mother's side) or F (father's side).



	Name	john	mary	bill	Tree status
*	John Smith	(SELF)	1000	900	100 People
*	Mary Jones	1000	(SELF)	.	50 People
*	Bill Johnson	900	.	(SELF)	No Trees
*	James Johnson	500	.	M	Locked
	Ann Peterson	400	F	.	Unlinked Tree
	Randy Jones	300	_	_	20 People
*	Angela Smith	300	F	M	2 People
	Kim Madison	250	.	M	No Trees
	Peter Wilson	200	F	.	123 People

Figure 11. HTML Output From The Program As Displayed By A Browser

SAS CODE SNIPPETS

I used several interesting techniques within the implementation of the read_matches4.sas program. In this section, I highlight several of these techniques with code snippets.

GATHERING FILE METADATA

The %file_metadata macro is used to create a SAS data set containing variables for each metadata value that is available. These will include filename, file size, and creation date and time. First, we use the DOPEN function to open the directory containing the files. DNUM tells us how many files there are, and DREAD will read a directory entry to obtain the filename. The second pass will use FOPEN on each filename, FOPTNUM for the number of metadata values available, FOPTNAME for the metadata name, and FINFO for the metadata value. When done, we transpose the data set so that the metadata names become the variable names. Note that VALIDVARNAME=ANY must be used here because the metadata names will contain blanks.

```
%macro file_metadata(dirname,slash,metadata_dataset);
data &metadata_dataset;
     length filename $256;
     rc  = filename('dfileref',"&dirname.");
     did = dopen('dfileref');
     nmems = dnum(did);
     do i=1 to nmems;
        filename = dread(did,i);
        output;
        end;
     rc = dclose(did);
     rc = filename('dfileref');
     keep filename;
     run;
data &metadata_dataset; set &metadata_dataset;
     fullqual = "&dirname.&slash."||filename;
     length opt $32 optval $256;
     do while(1);
        fid = 0;
        filename_rc = filename('myfile',fullqual);
        put filename_rc= fullqual=;
        if filename_rc ne 0 then leave;
        fid = fopen('myfile');
        put fid=;
        if fid = 0 then leave;
        n_metadata = foptnum(fid);
        do i=1 to n_metadata;
           opt = foptname(fid,i);
           if opt = "&opt_filename." then continue;
           optval = finfo(fid,opt);
           output;
           end;
        leave;
        end;
     if fid ne 0 then rc = fclose(fid);
     if filename_rc = 0 then rc = filename('myfile');
     keep filename opt optval;
     run;
proc transpose data=&metadata_dataset out=&metadata_dataset(drop=_NAME_);
by filename notsorted;
     var optval;
     id opt;
     run;
%mend;
```

An example of what the observations look like:

```
filename=rick_mother_matches.txt 'File Size (bytes)'n=21
 'Last Modified'n=03Feb2021:14:04:10
filename=rick_rick_matches.csv 'File Size (bytes)'n=22
 'Last Modified'n=03Feb2021:14:04:33
filename=rick_rick_matches.txt 'File Size (bytes)'n=20
 'Last Modified'n=03Feb2021:14:03:55
```

READING HTML DATA TO CREATE A CSV

With this SAS code snippet, I demonstrate a method I described in my 2009 SAS Global Forum paper (see References section for more information). RECFM=F and LRECL=&filesize. is used so that the entire HTML stream is read as a single record. The @'…' operator is used in the INPUT statement to locate specific HTML tags. And the GUID values, which are part of HREF addresses, are extracted.

```
data _null_; infile "&dirname.&slash.&filename."
             recfm=f lrecl=&filesize. end=eof column=c missover;
             file "&dirname.&slash.&part1._&part2._matches.csv";
     …
     put "name,guid,&part2,tree_status";
     input @'batBeacon' @;
     do while(c < &filesize.);
        input @'with%2F' @;
        if c < &filesize.
           then input guid $char36. @;
        …
        end;
```

We can use the file names and file sizes from the %file_metadata macro described above.

READING CSV FILES AND MERGING THEM

The CSV files that are previously generated in the program are read via this macro. PROC IMPORT is used to read each CSV file. The first time the macro is called, the data set ALL is created. For all subsequent times the previous version of ALL is merged. The GUID is used as the key. We need to ensure that the name is also added so it is included in the BY statement.

Once all data sets are merged, we will have unique variable names corresponding to the centimorgan counts of each. From our example in figure 7 above, the variable names would be john, mary, and bill.

```
%macro read_csv_and_merge(filename,first);
proc import file="&filename." dbms=csv out=temp replace; run;
data temp; set temp; guid=upcase(guid); run;
proc sort; by guid name; run;
%if &first %then %do;
data all; set temp; run;
%end; %else %do;
data all; merge all temp(drop=tree_status); by guid name; run;
%end;
%mend read_csv_and_merge;
```

COMPUTING CRC

The CRC is computed using the HASHING function. This function was introduced in 9.4m6 of the SAS System. Prior to 9.4m6, one would use the CRCXX1 function. I discussed these functions in my 2020 SAS Global Forum paper, which is in cited in the References section below.

```
crc = crcxx1(strip(guid)); /* Prior to 9.4m6, returns an integer */
crc = hashing('crc32',strip(guid)); /* As of 9.4m6, returns hex */
```

USING A FORMAT WITH THE CRC

The MATCHES data set has a set of numeric variables which are the centimorgan match values, or just the special missing value .Y if there is a match but the centimorgan count is unknown. This code uses an old-style array with the _NUMERIC_ shorthand to process all numeric variables. The VNAME function obtains the variable name, which will be the first name of the matching person. The CATS function continuously concatenates the data into a single label. The CRC is the key corresponding to the label.

```
data temp; set matches;
     fmtname='$cm';
     array cM _numeric_; /* john mary bill ... */
     start=crc;
     length label $100;
     label=' ';
     do over cM;
        name=vname(cM);
        if int(cM) ne cM then cM=.y;
        label=cats(label,',',vname(cM),':',cM);
        end;
     label=substr(label,2);
     keep fmtname start label;
     run;

proc format cntlin=temp; run;
```

The format is used in the following code. If brackets appear in the record, the CRC will be within the brackets. The CRC is extracted, and is replaced with the label from the format.

```
…
     i = index(_infile_,'[');
     if i then do;
        length part1 crc part3 $200;
        part1 = scan(_infile_,1,'[]');
        crc = scan(_infile_,2,'[]');
        part3 = scan(_infile_,3,'[]');
        _infile_ = trim(part1)
                   ||' ['||trim(put(crc,$matchinfo.))
                   ||']'||part3;
        output;
        end;
     put _infile_;
```

Extracting lines from figures 9 and 10 above, we can see how the _infile_ replacement appears:

```
Angela Smith 1948- [6784051D]
Angela Smith 1948- [john:300,bill:Y,mary:Y]
```

CONCLUSION

I have had a good deal of success using read_matches4.sas with Ancestry.com data to assist many people with determining unknown relatives. I have helped to reunite family members and bring closure to many peoples' lives. And I hope to continue using and improving on the technology to continue to assist more people.

REFERENCES

DNA Painter. https://dnapainter.com/tools/sharedcmv4

Langston, Richard D. 2009. "Creating SAS® Data Sets from HTML Table Definitions", Proceedings of the SAS Global Forum 2009 Conference. Available at http://support.sas.com/resources/papers/proceedings09/052-2009.pdf .

Langston, Richard D. 2020. "Slinging Hash: The HASHING Functions available in SAS®", Proceedings of the SAS Global Forum 2020 Conference. Available at https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2020/4838-2020.pdf .

Leeds, Dana. "The Leeds Method". https://www.danaleeds.com/tag/leeds-method/

Wikipedia. "Universal Identifier".https://en.wikipedia.org/wiki/Universally_unique_identifier

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

RickLangston1955@gmail.com