Grouping Large Datasets: Supervised & Unsupervised Leeds Method

James Sims PhD

Self-published online: Wed Oct 17 11:59:05 2018

Abstract

This article describes the behavior of the genetic genealogy technique known as the Leeds Method[1] when the size of the DNA match table increases past a few dozen matches, and when the original parameters of the technique are modified. In this study, a reproducible implementation of the Leeds Method in the R statistical programming environment was used to analyze the author's matches at AncestryDNA[2].

Introduction

Genetic genealogists face a challenging task when they try to properly categorize autosomal DNA matches as being due to DNA inherited from specific ancestors. The Leeds Method, developed by Dana Stewart Leeds, is a method originally designed to summarize predicted second cousin (2C) and third cousin (3C) matches at AncestryDNA by creating match groups. There are compelling reasons to use this method as originally described when *beginning* to assess how DNA matches are related to a test-taker as described previously by the author, cited above. However, for this study, a computer-based implementation of the Leeds Method in a programming language was a requirement for reasons of reproducibility and for relative ease of data analysis.

The behavior of the Leeds Method with large data sets has not been previously described. In this study, data sets of up to 2,571 matches were grouped by the Leeds Method. The number of groups and the size of the groups created by the method were assessed as a function of the size of the match list. The position of certain matches in the match table, who have more than one genealogical relationship with the test-taker, are expected to have effects on the number of groups created and the size of groups, especially if such matches are near the top of the match table. This is a consequence of the sequential row by row, top down nature of the method. This study reports on the positional effects of a pair of DNA matches for people who each have two genealogical relationships to the test-taker (the author), namely those of maternal 2C and maternal second cousin once removed (2C1R).

Data sources

The data sources from AncestryDNA and the methods used to obtain them for this study were the same sources and methods as described previously by the author. The .csv format match file and the .csv format in-common-with match file were imported into the free version of the R statistical programming environment[3] using the free version of RStudio[4], an integrated development environment for R. These data sources are not included in the GitHub repository with this article due to privacy considerations[5].

There were 62794 rows of data, one row per match, in the matches file. There were 584404 rows of data in the in-common-with file.

[1] Dana Stewart Leeds, Dana Leeds blog, accessed September 25, 2018

[2] James Sims, pub_leeds repository, GitHub, accessed September 24, 2018

[3] R Core Team (2018). R: A language and environment for statistical computing.R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

[4] RStudio Team (2016). RStudio: Integrated Development for R. RStudio, Inc., Boston, MA URL http://www.rstudio.com/

[5] James Sims, pub_leeds_4C repository, GitHub, accessed September 24, 2018

Applying Unsupervised Leeds to Large Datasets

Leeds' original implementation of the grouping method assumed no prior knowledge of the matches being grouped. In this study, the term unsupervised is used to describe that version of the Leeds Method and its lack of assumptions or knowledge about the matches, and makes no changes to the match table order. Leeds applied the method only to 2C and 3C matches originally. In this study, the match table was expanded to include matches that are predicted 2C, 3C and 4C cousins before applying the unsupervised Leeds Method. Because AncestryDNA requires in-common-with matches (shared matches) to be at least predicted 4C matches of the test-taker and the in-common-with match, predicted distant cousins were excluded. As described by Leeds, a second filter was applied to remove any matches that shared 400 cM or more DNA with the author.

After applying these filters, there were 2571 rows of data, one row per match. The filtered matches appeared to be sorted by shared cM values in descending order. A descending cM value order was enforced on the filtered matches by sorting the data in descending order in R. Looking at the filtered list of matches, the predicted cousin frequencies were as follows.

```
##
## FOURTH_COUSIN SECOND_COUSIN  THIRD_COUSIN
##          2529             3            39
```

The author's filtered matches were used to generate data sets of increasing size beginning with the first (top) row of the filtered matches being the first row of *each* data set. The match table sizes (matches, rows) chosen for this purpose represent seven doublings of the match table size plus a data set containing all of the filtered matches. Those data set sizes are shown in the output of R code below.

```
## [1]   16   32   64  128  256  512 1024 2048 2571
```

The following table shows the number of groups created by unsupervised Leeds as the match table size increases.

```
##   data_size num_groups
## 1        16          4
## 2        32          8
## 3        64         10
## 4       128         20
## 5       256         39
## 6       512         67
## 7      1024        112
## 8      2048        212
## 9      2571        269
```

The growth in the total number of match groups as a function of table size is also plotted in Fig. 1.

The growth of the first eight groups as a function of increasing match table size is shown in Fig. 2. In the R implementation of the grouping procedure, groups are given names rather than colors, and those names begin with the letter G, followed by a number. In this study, the line colors for groups G1-G8 were assigned automatically by R and have no significance other than the ease of presentation of the data in a single graph. In this labeling scheme, the G1 group was created before the G2 group, G2 before G3, etc. by unsupervised Leeds.

One group, G1, grew much larger and more quickly than the other first eight groups as shown in Fig. 2. Group G1 was created based on the first match (top row) of the match table. This match shares the most DNA with the author of any match in the match table. In the unsupervised Leeds Method grouping, we either know nothing about the match's relationship to the test-taker (in this case, the author) or we ignore what we know. Here, the case is the latter. This match is related to the author two ways: as a maternal 2C and as a maternal 2C1R.

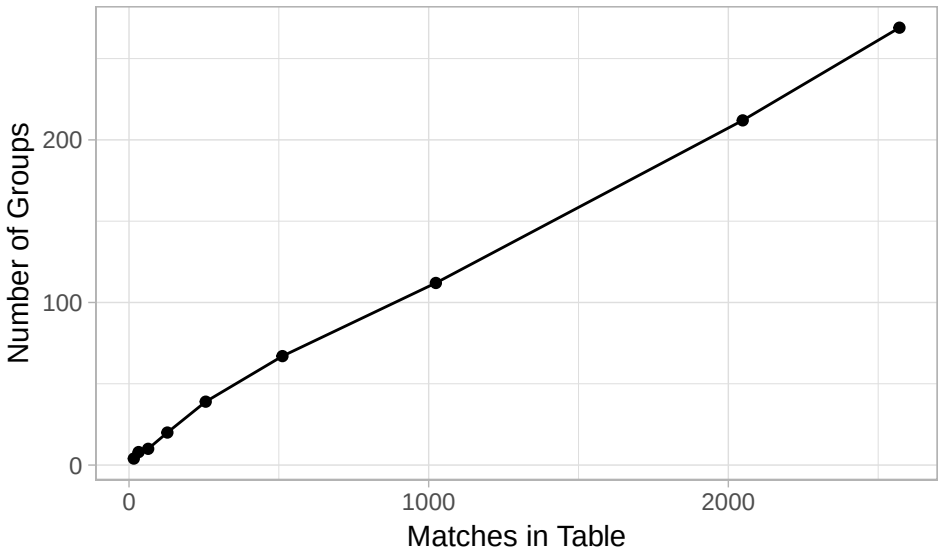

Figure 1: Increase in number of match groups as a function of table size for unsupervised Leeds.

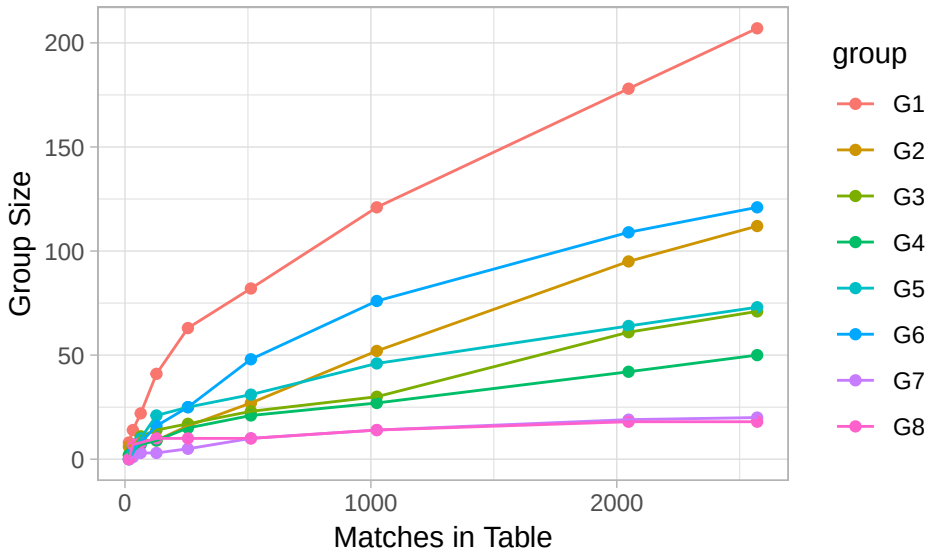

Figure 2: Increase in size of first eight match groups as a function of table size for unsupervised Leeds.

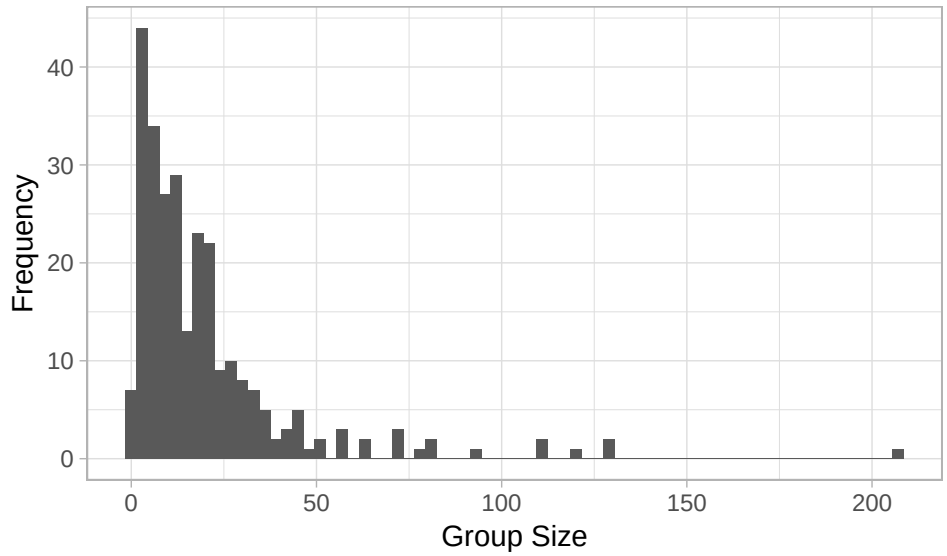

Figure 3: Distribution of match group size when the match table size is the largest for unsupervised Leeds.

The table shown in the R code output below summarizes the groups created by the unsupervised Leeds Method as the size of the match table size increases.

```
## # A tibble: 9 x 7
##   tsize groups  gmin  gmax gmean gmedian sd
##   <dbl>  <int> <dbl> <dbl> <chr> <chr>   <chr>
## 1    16      4     2     8 4.50  4.00    3.00
## 2    32      8     1    14 5.90  5.50    3.90
## 3    64     10     3    22 8.90  8.00    5.20
## 4   128     20     2    41 10.30 8.50    8.80
## 5   256     39     1    63 10.60 7.00    11.30
## 6   512     67     1    82 13.10 9.00    13.70
## 7  1024    112     1   121 16.20 11.00   18.00
## 8  2048    212     1   178 18.90 11.00   22.90
## 9  2571    269     1   207 20.00 12.00   24.20
```

In this table, *tsize* is the number of matches in the match table; *groups* is the number of groups produced for a given table size; *gmin* is the minimum size (number of shared matches) for the groups created; *gmax* is the maximum size for the groups created; *gmean* gives the arithmetic average size of the groups created; *gmedian* gives the value for the median group size, that is, the size of the group that half the groups are below and half of the groups are above; and *sd* is the standard deviation of the group sizes. When the mean (average) group size does not equal the median group size, this indicates the data are skewed (shift left or shift right) compared to what mathematicians call the normal distribution.

The skewed shape of the group size distribution is confirmed in the histogram showing the distribution of group sizes in Fig. 3. This is the general shape of the Poisson distribution, which is the distribution that describes collections of random events that ocurr independently of each other[6].

Applying Supervised Leeds to Large Datasets

In the unsupervised Leeds Method, group G1 grew faster and attained a much larger size compared to the other first eight groups as shown in Fig. 2 and Fig. 3. G1 was created based on who shares DNA with the

[6]Wikipedia, Poisson distribution, https://en.wikipedia.org/wiki/Poisson_distribution, accessed October 1, 2018

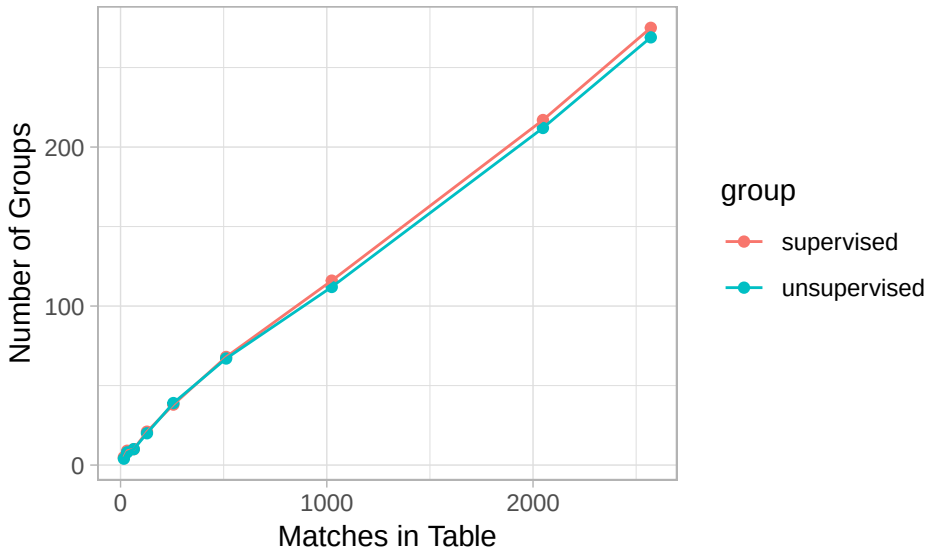

Figure 4: Comparison in the rate and extent of match group growth for two versions of Leeds.

first match in the match table. G1 contained 207 shared matches. This match and the second match in the match table are 1Cs relative to each other, and both of them each have a 2C and a 2C1R genealogical relationship with the author due to cousin marriages in the author's maternal lines. To test the effect matches 1 and 2 have on the grouping, as a first approach a slice/flip/stack methodology was use. For the slice, the top two rows of each data set (always the same two matches) are sliced from the top of the table. For the flip, the slice of two rows is resorted into ascending cM order. And for the stack, the flipped two rows are appended to the bottom of the match table.

The output of the R code chunk below is a table comparing the total number of groups created for supervised and unsupervised Leeds as a function of match table size. The number of groups created for match table sizes up to 512 were very similar. As the match table grew larger, supervised Leeds produced a few more groups with the difference being an additional six groups for the largest data set.

```
##   data_size supervised unsupervised
## 1        16          5            4
## 2        32          9            8
## 3        64         10           10
## 4       128         21           20
## 5       256         38           39
## 6       512         68           67
## 7      1024        116          112
## 8      2048        217          212
## 9      2571        275          269
```

Fig. 4 compares the growth in the total number of groups created by Leeds for the supervised and unsupervised versions of the method. The rate of growth was very similar for match table sizes up to 512 matches. The supervised version of Leeds produced a few more groups than the unsupervised version did when the match table was larger.

Fig. 5 shows the rate and extent of growth of the first eight groups created by supervised Leeds. The group names G1-G8 for supervised Leeds are not directly comparable to those shown in Fig. 2 for unsupervised Leeds. For example, in unsupervised Leeds, group G1 was based on a match that is maternal for the author, but for supervised Leeds, G1 is based on a paternal match for the author. This is due to moving the first two matches in unsupervised Leeds to the bottom of the match table for supervised Leeds and on the relative placement of other matches in the table.

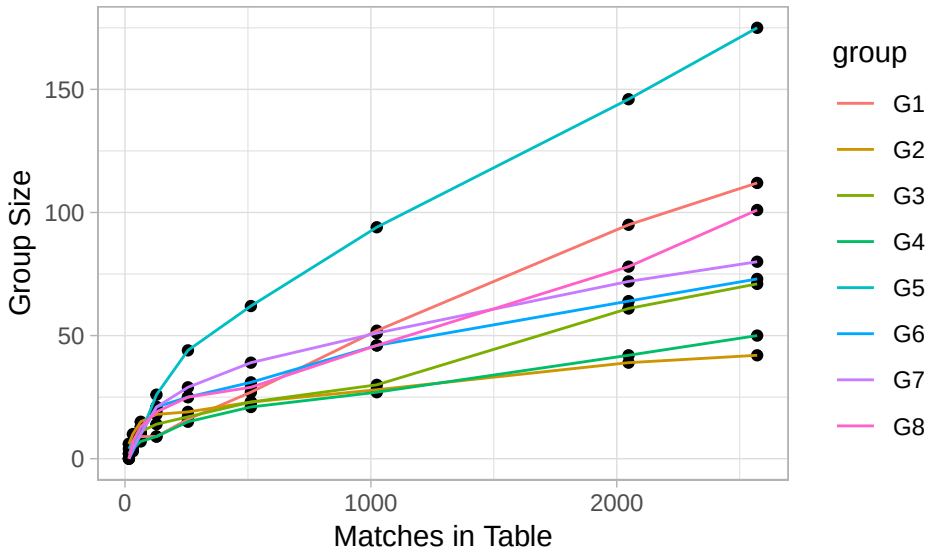

Figure 5: Growth of first eight match groups as a function of match table size for supervised Leeds.

As shown in the output of R code below for the supervised method, there were modest reductions in mean group size and in the standard deviation of the group size for supervised versus unsupervised Leeds. The was a significant decrease in the maximum size of the largest group for supervised Leeds compared to unsupervised Leeds.

```
## # A tibble: 9 x 7
##   tsize groups  gmin  gmax gmean gmedian sd
##   <dbl>  <int> <dbl> <dbl> <chr> <chr>   <chr>
## 1    16      5     2     6 4.0   4.0     2.0
## 2    32      9     1    10 5.3   5.0     2.7
## 3    64     10     3    15 9.3   10.0    3.6
## 4   128     21     2    26 9.5   6.0     7.2
## 5   256     38     1    44 9.8   6.0     9.2
## 6   512     68     1    62 12.5  9.0     11.4
## 7  1024    116     1    94 15.7  12.0    15.4
## 8  2048    217     1   146 18.2  11.0    21.0
## 9  2571    275     1   175 19.4  12.0    22.7
```

The distribution of group sizes for supervised Leeds is shown in Fig. 6. This distribution is very similar to that of unsupervised Leeds as shown in Fig. 3. To determine the magnitude of the effect of supervision on the matches moved to the bottom of the table (special cases), the number of groups for the the match sharing the most DNA with the test-taker was examined. The data for these comparisons is shown in the R code table below.

Leeds	Number of Groups	Group Identity (Gn)
unsupervised	1	1
supervised	17	2, 5, 7, 8, 20, 21, 34, 46, 50, 74, 101, 107, 114, 127, 132, 183, 193

For the match sharing the most DNA with the test-taker, that match was part of 17 match groups in supervised Leeds compared to just 1 match group in unsupervised Leeds.

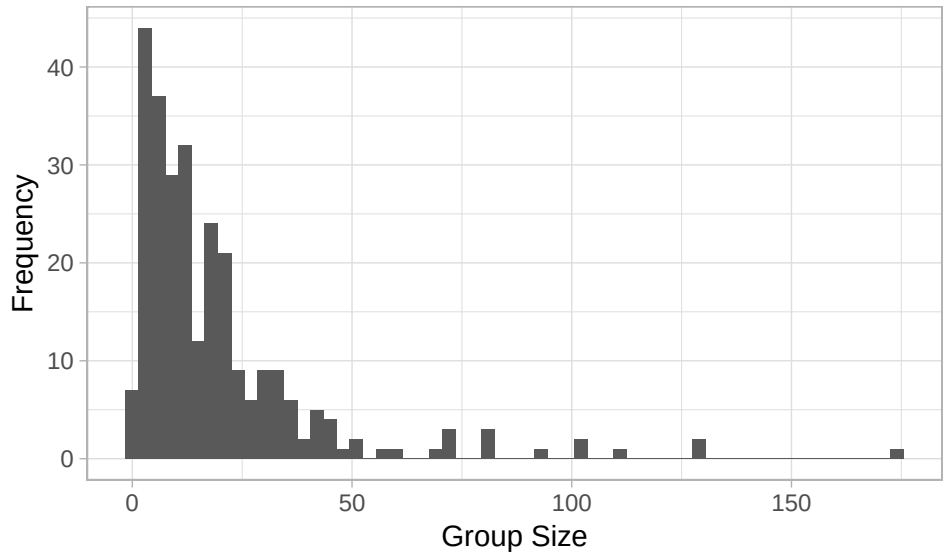

Figure 6: Distribution of match group size when the match table size is the largest for supervised Leeds.

Discussion

In our view, the Leeds Method as originally developed is an excellent tool to *begin* understanding predicted 2C and 3C cousin matches. It is well suited to the needs of beginning genealogists and those seeking birth families when the paper trail is very thin or nil. If one wants to apply the method to large groups of matches, say more than a few dozen, the method suffers from two problems. First, it becomes tedious. Experienced genealogists often tolerate a large amount of tedium in pursuit of insights, but inexperienced genealogists often have yet to develop their long game skills and may likely abandon tedious tasks. Second, it is easy to make mistakes applying the method and accuracy checking is also tedious. A computer-based algorithmic implementation of the method has the potential to deal effectively with both these issues. A well constructed algorithmic implementation will also be reproducible: given the same two data files, namely a match list and an in-common-with match list, and the same software, the algorithm will produce the same results every time.

The supervised version of Leeds implemented in this study may offer advantages which are not easy to fully quantify at this time because genealogical research on the match list is incomplete. However, the effect of supervision on the matches that share the most DNA with the test-taker can be dramatic. For example, for the match table with a row size of 2571 in this study, the match occupying the first row in the unsupervised version of Leeds was a member of 1 group. This is the same match that occupies the last row of the match table with 2571 rows in supervised Leeds. In supervised Leeds, that match was a member of 17 groups!

It is hard to overestimate the importance of how helpful categorized close matches are when trying to understand so-called more distant stranger matches. This example shows that valuable information may be overlooked if an unsupervised approach is taken and there are 2C matches in the match table with more than one *recent* genealogical relationship to the test-taker.

The author makes the following recommendations. First, use a supervised Leeds approach when appropriate. Second, use the slice/flip/stack method described here for matches less than 400 cM when you know some matches have more than one *recent* genealogical relationship with the test-taker. Third, take an all matches or full stack approach. Append all matches sharing 400 cM or more to the bottom of the match table with the largest shared cM values at the bottom of the table. See Fig. 7. A full stack table so constructed will have a large number of matches sharing less than 400 cM with the test-taker at the top sorted with cM values high to low, followed by matches with more than one known *recent* genealogical relationship to the test taker sorted with cM values low to high, followed by the matches that share 400 cM or more sorted with cM values low to high. A full stack approach should help the test-taker understand how their closest family members

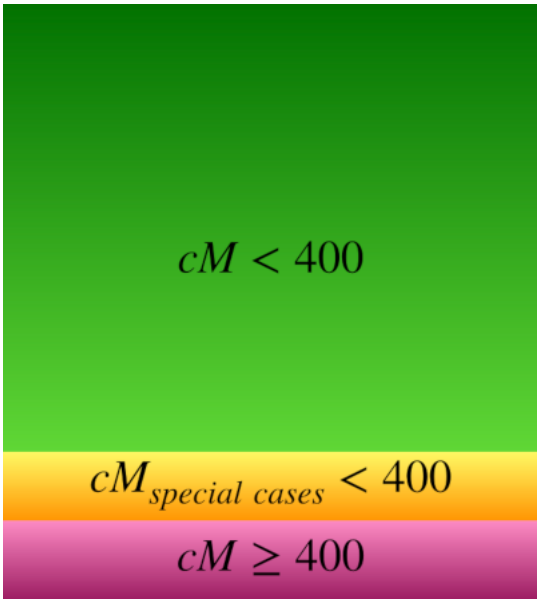

Figure 7: Full stack match table prepared for supervised Leeds. Gradient shows how matches are sorted, lighter is lower cM value matches. Colors show different portions of the table.

share DNA with the more distant DNA cousin matches. At some point in the future, if you need to add new matches, add them to the bottom of the table to preserve the integrety of the inital implementation.

To be clear, the results in this study are the results of applying the Leeds Method to the matches of only one person, namely the author. Although the methods used in this study should be applicable to most people's matches at AncestryDNA, and reproducible, a large variation in the number of match groups created by the method and the size of the match groups is expected for large data sets including 4C matches.

The vast majority of genealogists and genetic genealogists are not programmers, and learning to use a programming language like R is very, very challenging. What is needed is for some enterprizing programmer to implement an automated Leeds Method tool along the line presented in this work. The author does not have the skill set to provide a web interface or a cross-platform stand-alone application for automated Leeds.

Software

DNAGedcom Client version 2.1.6 (2.18) for Mac was used to download data from the author's account at AncestryDNA. The free R version 3.5.1 (2018-07-02) was used for this analysis. The code was developed in RStudio version 1.1.456. The tidyverse package version 1.2.1 was used to make coding easier and more readable for humans[7]. This report was produced within the RStudio integrated development environment using rmarkdown version 1.10 and the knitr package version 1.2. and its dependencies. MacTeX-2018 was used for pdf output on a 2017 MacBook Pro running macOS version 10.13.6 (17G65), which is commonly called High Sierra.

[7]Hadley Wickham, Tidyverse.org, tidyverse version 1.2.1