MATH 459 Final Project Report

Imputing Missing Values in Family-Associated Genomic Data

Hayden Brown
Binghamton University
Binghamton, USA
hbrown10@binghamton.edu

Bryan Edmonson
Binghamton University
Binghamton, USA
bryanedmonson25@gmail.com

ACM Reference Format:
Hayden Brown and Bryan Edmonson. 2019. MATH 459 Final Project Report: Imputing Missing Values in Family-Associated Genomic Data. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Data imputation for missing values is a pressing topic in research today, having been an important topic in medical research since at least 1999.[1] Being able to accurately guess the values of missing fields is incredibly valuable for expanding or restoring data sets. For example, in 2016 a set of data with imputed genotypes was used to find significant association of the passing of Alzheimer's disease with the genes OSBPL6, PTPRG, and PDCL3.[2] Missing value imputation can bridge generational gaps and glean insights into the way traits are passed from parent to child.

1.1 Problem of Interest

For our final project, we set out to build a pipeline that imputed missing values in family-based genomic data. This is a topic of strong interest for GWAS researchers, as not all genetic family-based data is complete for every genome instance across each family; with all missing values imputed, further analysis can proceed with the complete data set. Our group was assigned the task of replicating missing value imputation methods similar to those of field researchers upon a dataset of our choice.

1.2 Dataset of Choice

Because our project focused specifically on the missing data imputation, we needed a slightly different GWAS dataset than what may normally be found in the field. In order to confirm our pipeline's success, we need to begin with a complete data set and then randomly generate our missing values. The original data set would then be a useful point of comparison for our results.

The dataset that we chose is simply named "Families," found as an example dataset in the R package snpStats. It contains family-associated data, wherein hundred of families have genetic inheritance information recorded for some undisclosed genetic quality or mutation. The set is not related to any major study that we can find evidence of, and its details in the documentation for snpStats can be found below.[3]

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 METHODOLOGY

In this section, we will detail our entire pipeline process, from data cleaning to how our pipeline functions.

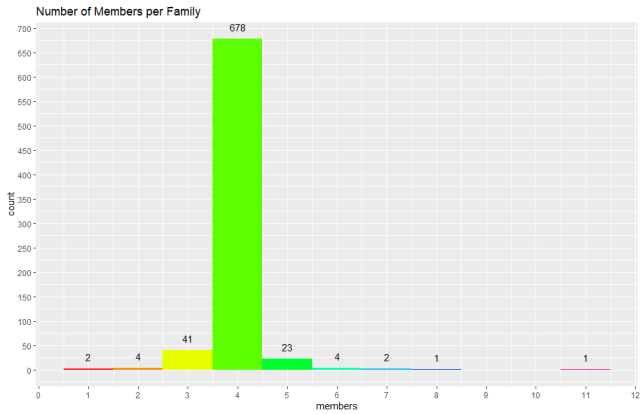
2.1 Software Used

Our pipeline was written in the R programming language using the RStudio IDE, with the help of a number of packages that will be cited below:

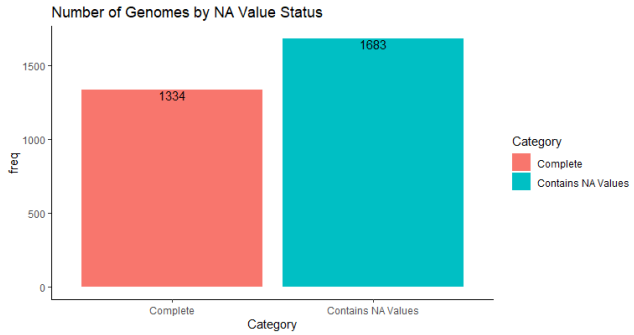
- ggplot2, for data visualization throughout this paper.
- grid and gridExtra, to help combine multiple plots of paired significance.
- lattice, for providing us with the robust densityplot function.
- mice, for providing the actual tools used to impute our missing data, along with some key visualizations of it.
- snpStats, which provided us with our data set and the tools to process it.
- tidyverse, for the ever-useful pipeline operator.
- VIM, for providing us with the robust aggression plot function, aggr.

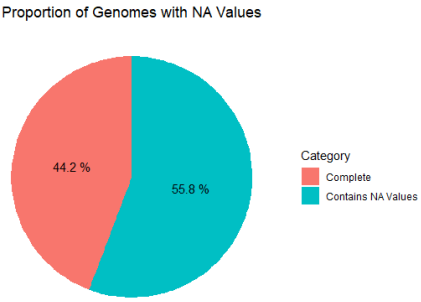
2.2 Data Exploration

The set contains 3017 genomes across 756 unique families. The number of members per family is not uniform across the families, and is distributed as follows. Families without at least 3 members (two parents and one child) were removed from the dataset.



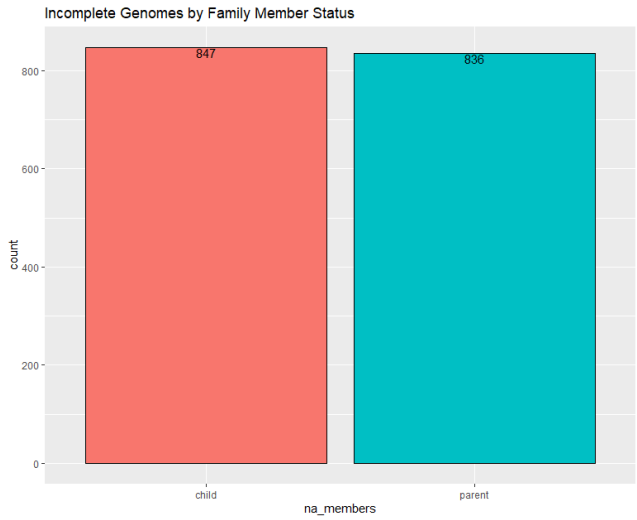
We found that a very large number of genomes in this data set contained NA values for various fields. In order to determine our pipeline's ability to impute data correctly, the original data set must be complete. As such, these genomes could not be used in our pipeline, but because of the additional constraint of the family structure, our team did some data cleaning with consideration of how those NA values were distributed in each family they appeared in.





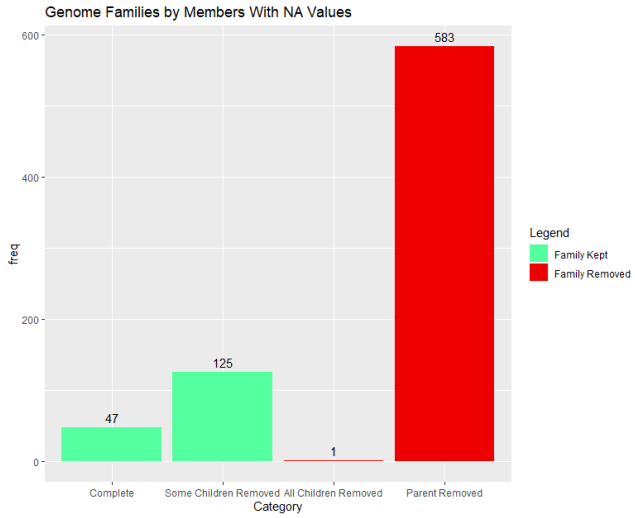
2.3 Data Cleaning

We evaluated which genomes with missing values were either parents or children in their family structure, as the member status had an impact on whether the rest of genome's family can remain in the data set.



In cases where one of at least two children had the NA values, only the child genome was removed from the set - the reasoning being that the rest of the family still had at both parents and at least one child with complete data. However, if all the children in a family had incomplete data, the family would be removed, as there would be only parental data remaining.

If either parent in a family had incomplete data, the entire family had to be removed, even if the rest were complete. This was unfortunate and led to a significant amount of data lost, but without both parents and at least one child per family in the set, there is no way to observe inter-generational genetic trends with any accuracy, and it may impact our imputation results later on.



Our final cleaned data set contained 508 genomes across 172 families, a figure that is smaller than the original set in genome count by a full 83.16%. This viable subsection is quite small in comparison to the full data set, but this cannot be worked around due to the limitations discussed prior.

2.4 Generating Random Missing Values

With our dataset trimmed to include only families with both parents and at least one child with complete genetic data, we started preparation for the actual missing data imputation. As stated prior, the missing data we want to impute is generated at random after we obtain our cleaned data set, so that we have a basis of comparison for accuracy. We decided on a removal proportion of 15%, and we randomly removed values across the cleaned data set in accordance with that figure.

2.5 Imputing Missing Values with MICE

Finally, we used the MICE package for R to impute the values of our data set. MICE stands for Multiple Imputation by Chained Equations, and imputes missing data using the method of its namesake. There is an assumption with MICE that the missing data in question is missing-at-random (MAR), and while we did use a random method to remove values previously, we will show that it's possible that it was not truly random. The package's observations on the missing data set, along with its imputation process and accuracy, are discussed in Results.

3 RESULTS

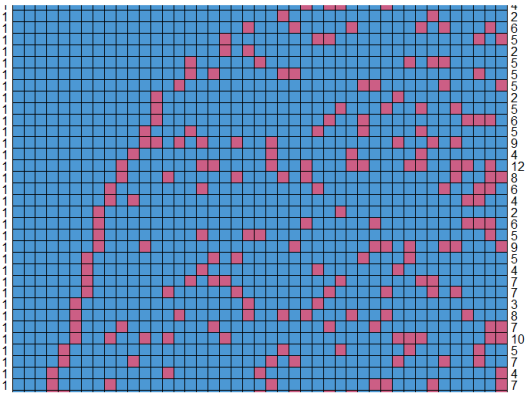
In this section, we will discuss the results of our MICE-based imputation pipeline, its limitations, and our final results as it pertains to the original goal of the project.

3.1 Randomness Disclaimer

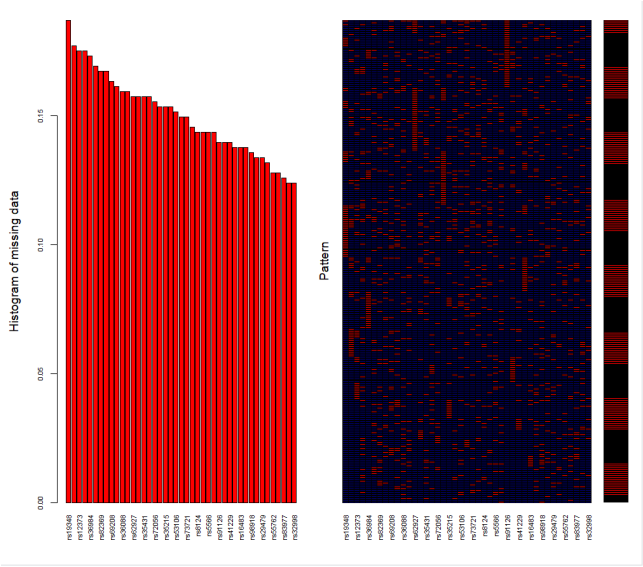
The random removal of data to create the set that was imputed, and the process of MICE itself via multiple imputation, both rely heavily on randomness and are thus not repeatable with exactly the same results. To escape this, we fixed RStudio's random seed in both instances during our work. The following results are thus only representative of a single attempt of the pipeline, and not comprehensive over multiple non-fixed runs.

3.2 Determining if Data is MAR

MICE's effectiveness as a method for missing value imputation heavily relies on the assumption that the data to be imputed is missing-at-random, or MAR. This generally just refers to the missing data not having any significant trends, such as only existing for one sample or factor, or a small cluster of either. To ensure our data would find success with MICE, we used the package's built-in pattern plot to visualize where in the set the missing values lie, and if there is any consistent pattern among them. There is a visible "boundary" in our set where the missing values begin to occur that is consistent throughout the data set, and that does indicate some level of patterned randomness. We believe that can be traced to the pseudorandom nature of the methods we used to remove data fields. However, the number of missing values per run and per factor were in accordance with the desired figure of 15%, so the data set was accepted and moved to the next step in our pipeline. The actual pattern plot was far too large to include in its full state, so instead the following plot represents the first 40 rows of data in the set.



Additionally, we want to ensure that no reads are missing values for more than 25% of their factors, because MICE effectiveness drops after that threshold. We generated a histogram of the missing data in the reads by proportion, and found that no reads reached that 25% mark, although a few were close.



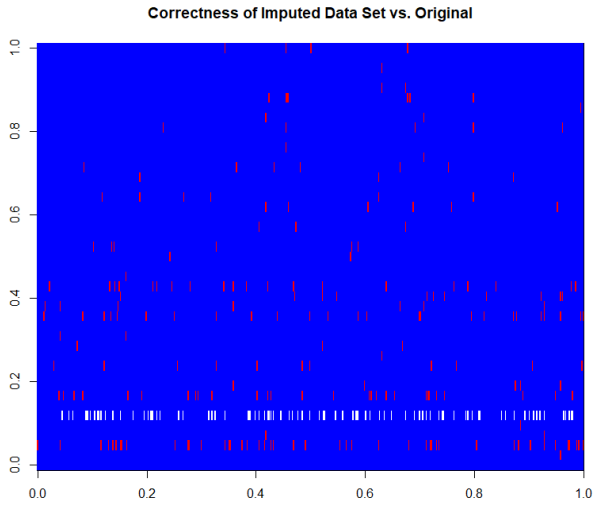
3.3 Imputation Choices

With our cursory data evaluation completed, we started picking the parameters for the imputation itself in MICE. In addition to a fixed random seed for reproducibility, MICE also takes two parameters "meth" and "m" for the imputation method and the number of times to impute each value for the multiple imputation. For our imputation method, we decided on the default method of "pmm," or predictive mean matching. PMM is more popular for quantitative imputation, and while our data was numeric, the domain of possible values (0, 1, or 2) defines our data set as categorical. However, MICE features no imputation methods that are strictly for use in

categorical methods with more than 2 groups, and instead recommends a handful of hybrid methods, including the aforementioned PMM. For the number of imputations, we chose m = 5, to reduce the overhead of the calculations. Our final selection method for the completed data was to select the mode of each imputation set for each value, to gather a consensus from the different imputation attempts. The density plots for each of the imputed factors across these 5 imputations is found below in Figure 1. It is worth noting that one of our 43 factors had no missing data, so it is not present in this chart.

3.4 Final Accuracy

Our final imputed set of data was then compared with the original set to determine the process's accuracy. On this particular random seed, the pipeline correctly imputed 3058 of a total of 3277 missing values, for an overall accuracy of 93.317%. The following chart is a comparison matrix that compares the values of the original data set with the imputed set. Discrepancies are marked red.



3.5 Discussion

The results we found indicate that our pipeline does in fact work as intended, with reasonable high accuracy as well. After the cleaning process, the data set used was relatively small in terms of the number of runs in comparison to other data sets, so the accuracy we were able to achieve bodes well for the effectiveness of the pipeline on both larger data sets and sets with a lower proportion of missing data.

Like stated above, this setup required multiple steps of randomness in both the missing data generation and multiple imputation stages, so it is certainly worth considering that our results are only definitively positive for our particular seeded attempt. Other random seeds may vary in effectiveness.

It is also worth considering that there are alternative parameters that could be tweaked to obtain potentially better results. In particular, MICE supports 4 other hybrid imputation methods that work for categorical variables with more than 2 groups, namely: "midastouch," a weighted PMM algorithm; "sample," a random sample from the observed values; "cart," an algorithm that uses classification trees; and "rf," a random forest imputation algorithm. It is unclear if others would have produced differing results, and this is worth looking into in the future. The number of imputation sets to generate could also be raised, given a smaller number of factors or less stringent time constraints.

4 CONCLUSION

In conclusion, we were able to effectively create a pipeline to solve the problem of interest with a high success rate. Outside of the concessions made earlier in Discussion, we are happy with our build of this code and believe it helped us understand the full steps needed for end-to-end data analysis.

We do think there are areas of potential improvement, however. We may have over-complicated the data cleaning portion of this project, as there was always the option of imputing the original data set to obtain a larger sample size. However, we were worried that with such a large amount of missing values, the imputation would be inaccurate, and later cause our own handmade imputation efforts to fall short in accuracy. The difference in overall performance, even for imputing values that are "faked," may have been negligible in retrospect.

We think that similar projects in the future should look into alternative data sets to avoid such problems. Starting with an incomplete dataset poses a number of conceptual challenges that may not be necessary or worth the effort. We still are grateful for snpStats providing the "Families" data for us, as many alternative data sets for research papers in the domain are not available for public use.

The individual contributions of each of our team's members will be submitted alongside this report.

REFERENCES

[1] Barnard Meng *Applications of multiple imputation in medical studies: from AIDS to NHANES* 1999
[2] Herold et al. *Family-based association analyses of imputed genotypes reveal genome-wide significant association of Alzheimer's disease with OSBPL6, PTPRG, and PDCL3* 2016
[3] David Clayton *'snpStats' Documentation* 2020

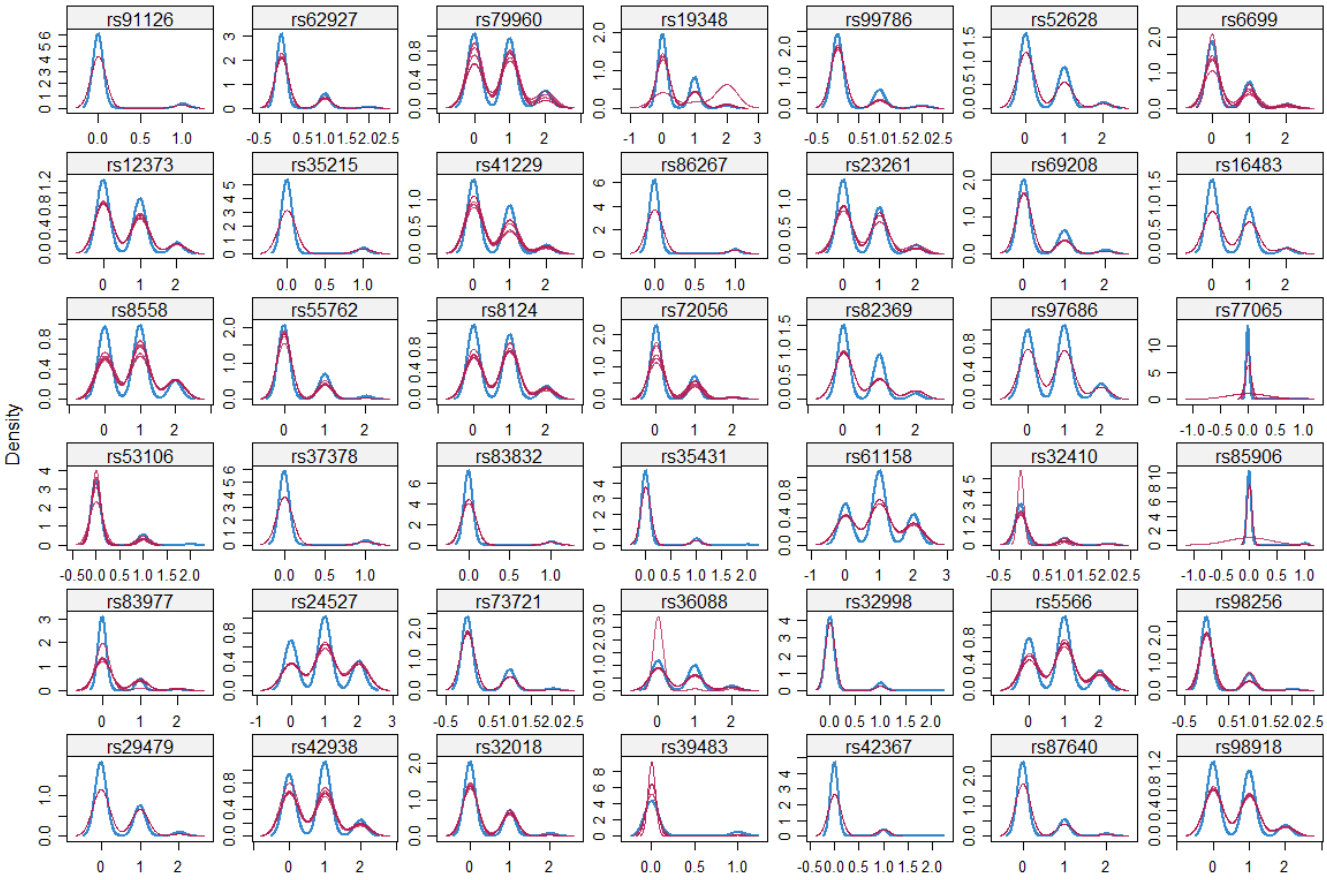


Figure 1: Density Plot of Imputed Factors