Addressing Missing Data in Viral Genetic Linkage Analysis Through Multiple Imputation and Subsampling-Based Likelihood Optimization

Addressing Missing Data in Viral Genetic Linkage Analysis Through Multiple Imputation and Subsampling-Based Likelihood Optimization

A thesis submitted by

Gabriel Erion

To

Applied Mathematics

in partial fulfillment of the honors requirements

for the degree of

Bachelor of Arts

Harvard College

Cambridge, Massachusetts

April 1, 2015

Abstract

This thesis addresses the intersection of two important areas in epidemiology and statistics: genetic linkage analysis and missing data methods, respectively. Genetic linkage analysis is a promising method in viral epidemiology which involves learning about transmission patterns by studying clusters of similar gene sequences. For example, similar sequences found in a pair of geographically distinct communities may imply disease transmission between the two locations. However, this analysis is sensitive to missing data, which can introduce substantial bias. This thesis presents a multiple-imputation approach which corrects for much, though not all, of the bias in genetic linkage analysis. It also introduces a novel resampling-based approach that generates a weighted distribution of complete datasets and is even more effective than imputation for reducing bias. This work highlights the importance of missing data in genetic linkage studies and presents ways to provide more accurate epidemiological information by correcting for missing data. The new resampling-based approach presented in this paper is also general enough to be applied to many types of missing-data problems involving complex datasets; such broader applications are a promising avenue for future research.[1]

[1] Code used in this thesis is available on Github at `https://github.com/gabeerion/thesis_2015`

Acknowledgements

There are many people without whom this thesis could not have been written. First, I must extend my most heartfelt gratitude to my advisor, Professor Victor De Gruttola. Over the past three years, working in his research group has immeasurably expanded both my skills and my confidence as a statistical researcher, and showed me the enduring joy of applying rigorous quantitative analysis to challenging problems. I would also like to thank Professor Joe Blitzstein not only for introducing me to statistics and teaching me many techniques which would be invaluable in this thesis, but also for discussing early drafts with me.

Many other members of my research group have provided guidance for which I am deeply grateful. Dr. Ravi Goyal has answered countless questions about Markov Chain Monte Carlo methods, and also provided my first introduction to network analysis. Dr. Vlad Novitsky guided me through the world of phylogenetic analysis and was incredibly knowledgeable about HIV biology and the challenges of working with large genetic datasets.

I would also like to thank the friends who selflessly took the time to read and provide comments on early versions of this thesis. Casey Fleeter, Robert Francis, Jake Freyer, Leo Guttmann, Dianna Hu, Ola Topczewska, and Paul Wei all read drafts with extraordinary care and thoughtfulness. They are what Patrick Rothfuss called "the kind of friends everyone wishes for but no one deserves."

Finally, and most importantly, a lifetime's worth of thanks to my parents, RL and Saila Erion. My gratitude for the curiosity and passion for learning you inspired in me is second only to my gratitude for twenty-one years of unconditional love and support. I love you and cannot thank you enough.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Inference in the presence of missing data is one of the most difficult problems in statistics, and a substantial body of literature addresses methods for performing such inference. One arena in which missing data becomes uniquely challenging is in the epidemiology of infectious disease. A powerful new idea in this field is that genetic sequence data of infectious microorganisms can be used to reconstruct information about the transmission networks along which diseases spread. This Viral Genetic Linkage (VGL) analysis relies on the idea that genetically similar microbes that infect different patients are likely to share a common evolutionary ancestor, implying that the infected patients are part of the same transmission chain. Epidemiologists have documented the ability of VGL to infer properties such as transmission rate, degree of disease exchange between communities, and even drug resistance. However, the fact that VGL relies not only on the distribution of individual genotype data but also on relationships between pairs of data points makes it particularly sensitive to missing data.

In this thesis, I present several increasingly sophisticated methods to correct for missingness in genetic sequence data, and demonstrate the ability of these methods to improve VGL analysis and epidemiological inference using a viral genetic dataset collected from HIV patients in Mochudi, Botswana. The first approach, multiple imputation, involves building a biological model to simulate missing gene sequence data. The second approach uses a resampling procedure to search for datasets that "fill in" missing gene sequences and maximize the likelihood of the incomplete dataset we did observe. Finally, I explore methods that generate a distribution of complete datasets, with higher-likelihood datasets given greater weight.

This thesis makes several contributions to the statistical and epidemiological literature. First, it demonstrates the bias that missing data can introduce in epidemiological studies of infectious disease, particularly VGL analysis. Second, it introduces a collection of novel, well-founded methods for general statistical inference in the presence of missing data. Finally, it demonstrates the effectiveness of these methods in the context of genetic sequence data and epidemiological inference by substantially reducing the bias in VGL analyses of HIV. My hope is that the availability of such methods will improve the tools policymakers and public health practitioners use in infectious disease policy and practice.[1]

1.1 HIV in Botswana

Botswana is one of the countries hardest-hit by the HIV epidemic and also one of the first African countries to implement a strong national response through testing and treatment. Historically, Botswana has had one of the highest HIV prevalences of any country; a 2011 study found prevalence to be 30.4% among women aged 15–49 [1]. Studies credit HIV/AIDS

[1] All code written for this thesis is publicly available on Github at `https://github.com/gabeerion/thesis_2015`

with a staggering 28-year decrease in life expectancy in Botswana between 1990 and 2006 [2]. One of the first major interventions undertaken by the government was a prevention of mother-to-child transmission (PMTCT) program begun between 1999 and 2001. Provision of free antiretroviral therapy (ART) began in 2002 and has been scaled nationwide [2]. The government of Botswana estimates that 67% of the total HIV-infected population is receiving antiretroviral therapy [1]. The government is continuing to work to increase access to ART, particularly among pregnant women, difficult-to-reach or stigmatized populations such as men who have sex with men (MSM), and individuals who have high risk for infection or for transmitting the disease.

The data used in this study was collected from Mochudi, Botswana, a community of over 44,000 people in southeastern Botswana. Since 2011, a pilot study in Mochudi has performed community-wide household surveys, provided HIV testing, and offered lab testing for viral load and CD4 count to HIV positive participants. The pilot study was itself a precursor to a larger planned study that would extend these services to 30 communities across Botswana and provide treatment for all patients with viral loads above 10,000 copies/mL [3]. The dataset analyzed in Chapter 2 represents 371 HIV subtype C gene sequences collected by the end of the first phase of the study in spring 2012. The data used in Chapters 3 and 4 consists of 1248 sequences collected between the beginning of the study in 2011 and summer 2014. The sequences are all from the V1C5 region of the *env* gene, aggregated as a multiple sequence alignment.

1.2 Viral Genetic Linkage Analysis

The core idea of Viral Genetic Linkage analysis is that genetic sequence similarity of infectious agents implies a common microbial ancestor. This allows us to draw various useful

epidemiological conclusions: for example, if many sequences in a community are genetically similar, the disease in question may be spreading quickly without time to accumulate mutations. If similar genetic sequences are found in geographically distinct communities, this may imply transmission between the two, and so on.

1.2.1 VGL Method

More formally, we can represent sequence data as an $n \times m$ matrix, where there are n patients in the population and a gene sequence of m sites is collected from each patient. We consider only datasets consisting of aligned sequences, which is reasonable considering that VGL is based on assessments of sequence similarity. Thus we might have an alignment A like the following:

$$A = \underbrace{\left.\begin{bmatrix} A & T & G & C & C & \dots \\ A & T & G & T & G & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ A & T & G & C & G & \dots \end{bmatrix}\right\} n \text{ sequences}}_{m \text{ sites}}$$

Each row of this alignment is a gene sequence. It's worth noting that this paper considers both nucleotide and amino acid sequences — any strings between which meaningful distance metrics can be constructed will suffice. In general, the strings have characters drawn from an alphabet Σ. The number of possible characters, or $|\Sigma|$, is usually 5 for nucleotide data (4 characters + 1 character to represent sequence gaps) and 21 for amino acid data (20 characters + 1 gap character.) For convenience, this section will exclusively consider nucleotide sequences. Given the alignment A, the row i represents the HIV gene sequence

collected from one patient. It can be written as

$$s_i = \begin{bmatrix} A & T & G & C & G & \dots \end{bmatrix}$$

The core step in VGL analysis is to form clusters of genetically similar sequences, which will form the basis for epidemiological inference. There are many ways these clusters can be defined. Phylogenetic methods that involve building a tree are a natural way to define clusters; branch length and preservation of proximity across bootstraps can be used to assign sequences to clusters [4–6]. Genetic distances between sequences can also be used to create clusters. One method is simple hierarchical clustering; another approach involves building a graph where nodes represent sequences and edges represent pairs of sequences that are separated by a distance below a given threshold [6]. Clusters can then be constructed from cliques or connected components in the graph. Previous work with VGL analysis in our group has focused on this last approach: to construct clusters we build a distance matrix and consider as clusters all connected components in the induced graph. It is worth noting that clustering methods such as k-means are not considered in this paper. While k-means is a simple and effective clustering method, it involves pre-specifying the number of clusters to be found. Since this number is itself useful information, it seems preferable to use biological knowledge about the genetic sequences in question to specify a reasonable genetic distance cutoff and see how many clusters naturally emerge.

For the purposes of assessing sequence similarity in the population, we construct a pairwise distance matrix between all sequences. Let the distance matrix D be constructed such that $D_{ij} = d(s_i, s_j)$ according to some distance metric between sequences. For example, if we use the Hamming distance (defined as $d(a, b) = \sum_{i=1}^{k} I(a_i \neq b_i)$ where a and b are strings of length k), then D_{ij} simply represents the number of sites in the alignment where

sequences s_i and s_j feature different nucleotides. Hamming distance is a very simple metric, and does not account for differential rates of mutation between different nucleotides or different frequencies of mutation at different sites in a sequence. Evolutionary biologists have developed many more complex metrics of genetic distance, which use known mutation frequencies to estimate the evolutionary time required to mutate one sequence into another. This study uses Hamming distance for simplicity; however, it's important to note that one advantage of VGL is that it can incorporate arbitrary distance metrics. A simple and efficient or complex and biologically realistic model can be used as desired, so long as it is possible to construct a pairwise distance matrix.

For example, if we have the very small alignment

$$A = \begin{bmatrix} A & T & G & C & C & G & G \\ A & T & G & T & C & G & G \\ A & T & G & C & C & G & G \\ A & T & G & T & G & G & G \\ A & T & G & A & A & G & G \end{bmatrix}$$

Then the resulting distance matrix is

$$D = \begin{bmatrix} 0 & 1 & 0 & 2 & 2 \\ 1 & 0 & 1 & 1 & 2 \\ 0 & 1 & 0 & 2 & 2 \\ 2 & 1 & 2 & 0 & 2 \\ 2 & 2 & 2 & 2 & 0 \end{bmatrix}$$

To form clusters, we need to choose a genetic threshold distance t below which we consider sequences similar enough to share a cluster. In general, one would like this distance to be chosen with some prior knowledge about expected genetic diversity of the infectious agent. If the maximum possible genetic distance is $d_{\max}$, sequences s_i and s_j cluster if $d(s_i, s_j) \leq td_{\max}$. In the case of Hamming distance, $d_{\max} = m$ because no sequence can differ from another at more than m sites. For illustration, if we choose the arbitrary threshold $t = 0.25$ so that the threshold distance is $tm = (0.25)(7) = 1.75$, then two sequences s_i and s_j are considered "linked" if $d(s_i, s_j) \leq tm = 1.75$. The adjacency matrix for the resulting graph, which will also be called the "clustering matrix", has entry i, j equal to 1 if s_i and s_j are linked and 0 otherwise. Formally, $M_{ij} = I(D_{ij} \leq tm)I(i \neq j) = I(D_{ij} \leq 1)I(i \neq j)$.

$$M = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Note that we remove self-loops for clarity; a "cluster of one" is considered trivial. The graph itself can be seen in Figure 1.1. If a cluster is defined as a clique, sequences (rows) 1, 2, and 3 form a cluster. If a cluster is defined as a connected component, as in this paper, sequences 1, 2, 3, and 4 form a cluster.

1.2.2 Application

If highly related clusters are interpreted as possible transmission partners, the properties of the clusters formed in VGL analysis can provide valuable epidemiological information

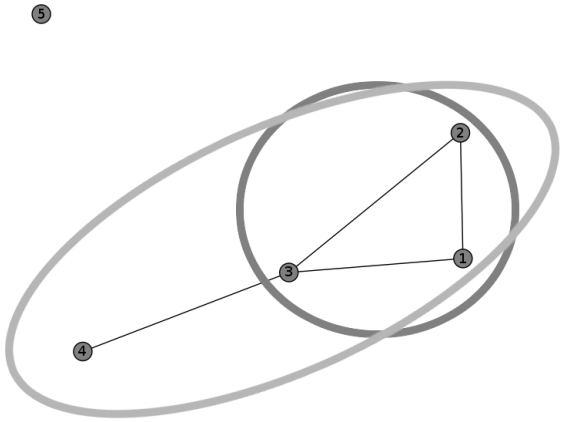

Figure 1.1: Graph corresponding to adjacency matrix M. If a cluster is defined as a clique, sequences 1, 2, and 3 form a cluster (dark grey). If a cluster is defined as a connected component, as in this paper, sequences 1, 2, 3, and 4 form a cluster (light grey).

[4, 7–9]. The literature surveyed here focuses primarily on HIV, both because it has been the focus of many efforts to leverage genetic information in public health and because studies focusing on HIV are most closely related to the work in this thesis.

Several studies use the size of sequence clusters to infer the number of neighbors of each patient in the transmission network; this approach also allows for direct estimation of the basic reproduction number R_0 [10–12].[2] Further, [12] shows that the distribution of cluster sizes can be used to estimate prevalence of HIV as well as R_0. When temporal data is available, the time between infections in a cluster can be used to estimate transmission rates (the number of transmissions per unit time) [13,14]. In addition, [14] incorporates data on infection stages to determine that transmission rates vary over the course of infection.

Viral genetic linkage analysis becomes especially useful, however, when clustering between and within communities and populations is analyzed. For example, [13] compares transmission rates estimated from epidemics occurring within heterosexual transmission networks to those estimated from MSM (men who have sex with men) networks, and finds that epidemics in heterosexual populations exhibit lower transmission rates. There is par-

[2]The basic reproduction number R_0 is the expected total number of new infections that a single infected individual will generate in an uninfected population.

ticularly strong public health interest in understanding which clusters new infections tend to fall into, with the hope that targeting these clusters for treatment can reduce the rate of new infections. Several papers have noted that new infections tend to cluster with other recent infections [4, 15]. Drug resistance is another characteristic of interest, and sequences featuring resistance mutations also tend to cluster together [7]. Epidemics in heterosexual, MSM, and IDU (intravenuous drug user) populations feature different avenues of transmission and different epidemiological characteristics; viral genetic linkage analysis has shown that sequences from any one of these subpopulations tend to cluster together [16]. However, clusters often contain both heterosexual and IDU patients, indicating that while the MSM epidemic is relatively isolated, cross-transmission occurs between the heterosexual and IDU epidemics. At a broader level, VGL has been used to assess transmission dynamics between communities. One study replicated the finding that heterosexual and MSM epidemics do not intermix, but did find clusters spanning multiple countries and noted evidence of transmission between populations in Nairobi and coastal Kenya [17]. A similar study using all publicly available HIV-1 polymerase sequences built a global map of viral genetic linkage between all countries and found multiple transmission clusters that crossed borders [18].

Finally, several studies attempt to use transmission networks constructed from closely related clusters to determine how to target interventions. One study noted that the distribution of cluster sizes in MSM epidemics is so skewed that randomly distributed interventions are unlikely to reach the small number of "superspreaders" driving the epidemic [6]. A later study attempted to address this problem by generating risk scores from a derived transmission network and demonstrated that this score was correlated with risky sexual behavior and risk of future transmission [19]. Simulations of interventions that targeted treatment

to individuals with high risk scores predicted significantly reduced HIV transmission across the whole network.

1.3 Impact of Missing Data on Genetic Linkage Analysis

The clusters studied in Viral Genetic Linkage Analysis become challenging to work with when data is missing because they depends on interactions between data points. Because of these dependencies, even data that is missing uniformly at random can induce bias in estimates. In particular, suppose that the property of interest, which we will call c, is defined as the proportion of sequences in the dataset that fall into any cluster. That is, how many sequences out of the total are close enough to some other sequence to fall below the genetic distance threshold? Because sequences and the distances between them are analyzed in a very high-dimensional space (for example, 1,000 dimensions define a 1,000-nucleotide sequence), it is hard to visualize how deletions affect the distances between sequences. However, the same principles apply to points in smaller spaces, like 2-dimensional Euclidean space, as well. Later chapters will demonstrate an empirically observed bias in c as sequences are deleted from HIV datasets (See Figures 2.1 and 2.2, as well as Table 2.1); this section will illustrate how missingness leads to biases in the distribution of distances by examining more easily visualized data in 2-dimensional space.

Figure 1.2 shows 128 points drawn at random from the unit square, that is with $x, y \in (0, 1)$. Half of the points are deleted uniformly at random in each subplot from left to right. As the number of points decreases (128, 64, 32, 16, 8), the average distance from each sequence to its nearest neighbor tends to increase (0.048, 0.044, 0.055, 0.082, 0.093) while

the proportion of points falling into a cluster tends to decrease (0.95, 0.86, 0.69, 0.63, 0.75). The threshold distance here is a Euclidean distance of 0.1. Note that though these are clear trends they are not monotonic. Occasionally, by random chance, the deleted points will be those which are far away from the rest; these deletions will decrease the average distance between points. Figure 1.3 shows histograms of the distance of each point to its nearest neighbor, which demonstrate that the distribution of minimum distances between points spreads out to larger and larger values as more points are deleted. In particular, the more points are deleted, the less probability mass remains on the left side of the threshold distance.

The overall proportion of clustering for a gene sequence alignment is calculated in the same way and can be interpreted as the probability mass of sequences that fall left of the threshold genetic distance, so it is vulnerable to the same mechanism of bias. As more sequences are deleted from the dataset, the distribution of distances between sequences becomes skewed toward larger values and less clusters emerge. Previous work has found that, in HIV testing studies, up to 40% of data is missing (minimum response rates are 60% among men and 70% among women) [20]. This illustrates the need for methods which can address the biases induced by missing sequence data.

1.4 Review of Missing Data Methods

This section will survey some traditionally used methods for addressing missing data and note their strengths and weaknesses for the Viral Genetic Linkage problem. Stef van Buuren notes in his text *Flexible Imputation of Missing Data* [21] that "[t]he standard approach to missing data is to delete them." However, he also notes that many more effective methods

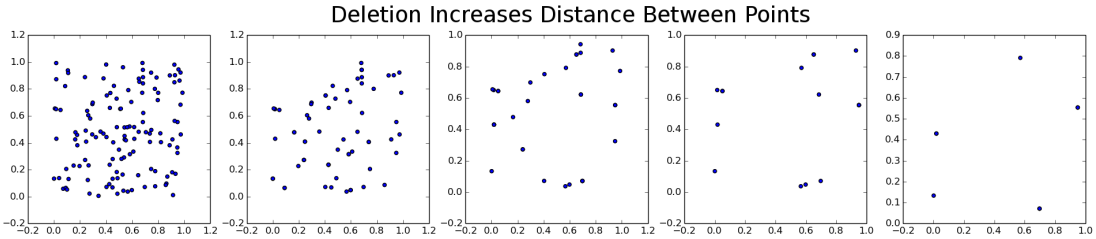

Figure 1.2: Plots of randomly generated points in 2-dimensional space, with $x, y \in (0, 1)$. This visualization shows that as points are deleted, the average distance between points increases. When distance metrics are used to define clusters, fewer clusters will be observed as points are deleted, even if deletions are uniformly random. The bias introduced in VGL sequence analysis is exactly the same but in a much higher-dimensional space.

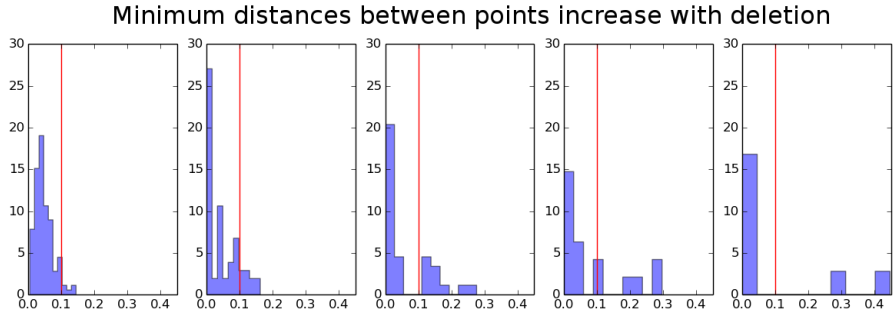

Figure 1.3: Histograms of the distribution of the distance from each random point in 2-dimensional space to its nearest neighbor for 128, 64, 32, 16, and 8 points (left to right). The distribution of these minimum distances starts out very skewed to the left, with most distances very small. As more points are deleted, the distribution spreads toward greater distances. The red line indicates the Euclidean distance below which points are defined to cluster together, analogous to genetic distance in VGL; note that, as points are deleted and the mass of the probability distribution spreads to the right, a smaller and smaller proportion of sequences will cluster together.

have developed for addressing missing data, from heuristics to rigorously justified alorithms.[3] Mechanisms of missingness and common methods for addressing missing data are discussed below.

1.4.1 Mechanisms of Missingness

It is possible to distinguish between three types of missing data, as discussed in [22]. Methods for addressing missingness will vary based on the mechanism by which data went missing.

- **Missing Completely at Random (MCAR):** In these cases, the probability that a particular data point from the full dataset goes missing is distributed uniformly, so that all points are equally likely to be missing. This is the least challenging scenario because ignoring the missing data still yields a dataset that is a representative, random sample from the population. However, as noted in the previous section, MCAR still results in bias when the statistic of interest is a function of the distances between data points in some space. Thus many methods that are generally assumed to work under MCAR will not always be suitable for VGL analysis.

- **Missing at Random (MAR):** When data are missing at random, the probability that a data point is missing depends on observed characteristics of the data but not on unobserved characteristics. For example, it is possible to categorize HIV patients into multiple demographic groups. If sequences from patients in different demographic groups have different probabilities of going missing, but all patients within a given group are equally likely to be missing, the data are MAR. This setting is more challenging because ignoring missing data does *not* result in a representative random sample. However, knowledge of which data are most likely to be missing makes it possible

[3] The text by van Buuren was a valuable reference in writing this section and provides more detail on all these methods.

to adjust for the missing data by stratifying before performing inference, reweighting existing data points or simulating new data. This is the case that applies when we know how many sequences in each demographic group are missing, but not what the sequences are.

- **Not Missing at Random (NMAR):** Data are NMAR if missingness depends on unobserved characteristics of the missing data; for example, if whether a sequence goes missing depends on properties of the actual sequence. Addressing missingness in this framework is extremely challenging, and is impossible without specifying a model for the process by which data goes missing. In the context of HIV, sequences would be NMAR if missingness depended on the distance between a sequence and its nearest neighbor, which is impossible to estimate without knowing the missing sequence. Though the NMAR setting is not explicitly studied in this thesis, so long as a mechanism for missingness can be specified, the subsampling-based methods presented in Chapters 3 and 4 can incorporate this mechanism to address NMAR data.

1.4.2 Methods for Addressing Missing Data

- **Complete case analysis** simply refers to eliminating missing data from analysis (only studying complete cases). It has the advantage of being very simple, but is problematic because it fails in all but the MCAR scenario. Complete case analysis can also reduce power because the size of the dataset is reduced.

- **Mean imputation** involves replacing all missing values of a variable with the observed mean of that variable and performing analysis on this "full data." It is worth noting that mean imputation biases the distribution of data by substantially increasing the probability mass placed on the mean. In addition, if the observed mean is biased (as

will occur in all but MCAR settings), mean imputation will reinforce the wrong mean. Mean imputation is particularly problematic in the sequence analysis setting: the most natural implementation would be to replace all missing sequences with the consensus sequence.[4] However, this causes particularly serious problems when properties of interest depend on genetic distance, as introducing many copies of the same consensus will result in a large number of zero genetic distances (in contrast, the datasets used here involve the variable *env* gene and have no zero distances). Thus, in VGL studies, mean imputation may be worse than complete case analysis.

- **Regression imputation** uses a regression model to predict values of the missing data based on the observed data, then performs analysis on the "full" dataset. This method works very well in the MCAR setting, and is also effective in MAR if the variables that determine missingness are included in the regression. This method does tend to unrealistically reduce variance of the data by imputing data along the regression line; however, this problem can be overcome by adding noise to the imputed values (stochastic regression imputation). More importantly, in the context of VGL analysis, it is difficult to generate new sequences based on observed ones with a simple regression; more sophisticated models are required. In addition, stochastic regression imputation draws noise from the observed distribution of residuals, which may not give an accurate assessment of the variance of estimates for arbitrary properties of interest.

- **Reweighting methods**, in particular inverse probability weighting, weight data points that were more likely to go missing under the missingness model more highly in the analysis. Reweighting methods are a very popular way to handle missing data, but

[4]The consensus sequence for a gene sequence alignment is defined as a sequence for which, at each site in the sequence, the nucleotide or amino acid present is the most common one in the alignment at that site.

some types of data cannot be reweighted in a straightforward manner. In particular, it is challenging to reweight genetic distance data between pairs of points. However, recent work in our research group has made progress on applying concepts from inverse probability weighting to sequence linkage analysis [23].

The techniques listed above do not constitute an exhaustive list; however, they do represent some of the most common approaches to handling missing data. Many of them are difficult to apply to VGL analysis or will yield biased estimates of the properties we are interested in studying. For this reason, the first method we present will be a very powerful technique not yet mentioned: multiple imputation.

1.5 Overview of Following Sections

The chapters proceed in roughly the same order in which research was performed and present increasingly sophisticated approaches to addressing the impact of missing data on VGL analysis. Chapter 2 presents and evaluates a multiple imputation approach for simulating missing data. Chapter 3 presents a subsampling-based method for scoring datasets that fill in the missing data. Chapter 4 presents methods for using this scoring technique to generate a distribution over possible full datasets. Finally, Chapter 5 discusses implications of the results and directions for future study.

Chapter 2

Multiple Imputation

One of the most intuitive approaches to dealing with missing data is to build a random model that can be used to simulate additional data until the dataset is "complete." Iterating this process multiple times to account for model variance and averaging across the results is known as multiple imputation, and is widely regarded as a gold standard technique for handling missing data. In this section, I demonstrate how a multiple imputation approach can reduce bias and improve estimates of variance in VGL analyses. Most of this section is based on work in my research group submitted for publication in [24].

2.1 Framework

Recall that we start with a matrix A representing a genetic sequence alignment of n sequences and m sites. The rows of this alignment are a proper subset of the rows of an unobserved, "true" full dataset which we'll call A^*. Let us rewrite n as n_o, for the number of observed sequences, and denote the number of missing (unobserved) sequences as n_m, so that the total number of sequences in A^* is $n_t = n_o + n_m$. The goal is to use the $n_o \times m$

alignment A to estimate a set of new *imputed* alignments $\{A'_1...A'_k\}$, each an $n_t \times m$ alignment. For a given A'_i, the first n_o rows should correspond exactly to A while the last n_m rows represent inferred sequences.

Suppose we are interested in estimating a property c of the true alignment A^*, and let f denote the function that calculates this property, so that $c = f(A^*)$.[1] Estimating c from the observed data, $c_{\text{obs}} = f(A)$ is the most obvious solution; we may even be able to estimate the variance $v_{\text{obs}} = \text{Var}(c)$ of this estimate by bootstrapping [25].[2] However, this estimator is biased for most properties of interest (examples were shown in Section 1.3) and bootstraps may give incorrect variance estimates. In the multiple imputation framework, we use the imputed alignments A'_i to obtain a more accurate estimate of c and its variance. In particular, we calculate $C_{\text{imp}} = \{c^1_{\text{imp}}...c^k_{\text{imp}}\} = \{f(A'_1)...f(A'_k)\}$ as well as $V_{\text{imp}} = \{v^1_{\text{imp}}...v^k_{\text{imp}}\} = \{\text{Var}(c^1_{\text{imp}})...\text{Var}(c^k_{\text{imp}})\}$. Then the multiple imputation estimate for c is just the mean of C_{imp}:

$$c_{\text{imp}} = \frac{\sum_{i=1}^{k} c^i_{\text{imp}}}{k}$$

The variance estimate is calculated as in [26]:

$$v_{\text{imp}} = \frac{\sum_{i=1}^{k} v^i_{\text{imp}}}{k} + \frac{k+1}{m}\text{Var}(C_{\text{imp}})$$

In [26] the first term is referred to as the "within-imputation variance", and represents the average variance of any individual estimate. The second term is the "between-imputation variance", and represents the variance of the vector of all estimates taken together.

[1] In all work in this paper, alignments A, A', A^* are matrices of characters or discrete integers representing those characters, and the properties c to be estimated are real numbers. While functions mapping, for example, from matrices to vectors or other objects are not considered in this paper, it is usually possible to incorporate them into this framework.

[2] Felsenstein first noted in his 1985 paper that it is possible to accurately assess the variance of statistics estimated from a phylogeny by bootstrapping. Given an alignment A, a large number of new bootstrapped alignments are generated by resampling m columns from A with replacement. The variance of the property of interest is calculated across the bootstrapped datasets.

The multiple imputation formulas given above allow for the estimation of arbitrary statistics and the construction of confidence intervals even when data is known to be missing, so long as it is possible to impute full datasets $\{A'_1 ... A'_k\}$.

The main remaining question is how to impute k datasets of $n_t \times m$ rows and columns from the observed data. If all that is observed is the $n_o \times m$ matrix A, we may not even know n_t. Not only are we unaware of the content of the missing rows, we do not even know how many rows there are! It is essential to have some information about how much data is missing and what kind of data it is. The approach adopted here makes use of two additional kinds of information to begin the imputation process, both acquired from census data. First, HIV prevalence statistics can be used to calculate the expected number of HIV patients in a population. This number is used as n_t, the observed number of sequences is n_o and $n_m = n_t - n_o$. Second, demographic information can be used to place patients into categories that provide more information about patterns of missingness. For example, if there are d different demographic categories in the population, then each sequence can be assigned a number between 1 and d representing which category it falls into. Thus, demographic data for the entire population can be represented as a vector T of length n_t where each entry is a patient's demographic category. Given this information, we know both the number and type of missing sequences. For example, if there are two demographic categories of interest, prevalence data may tell us not only that there are n_m missing sequences but that twice as many of these sequences fall into category 1 as into category 2. This allows us to construct a vector F of probabilities, with length d, where the ith entry represents the proportion of the population falling into category i. Thus, the available information for imputation consists of an alignment A, a number of missing sequences n_m, and perhaps also demographic information such as observed demographic categories T_{obs} and census-derived

demographic proportions F. The next section will show how to use this information to simulate the n_m missing sequences.

2.2 Multiple Imputation Method

Given the information above, a model of evolutionary mutations can be used to impute the n_m missing genetic sequences. The key observation enabling imputation is that each sequence in the population has a nearest neighbor, as measured by genetic distance. In terms of the pairwise genetic distance matrix D, the nearest neighbor for sequence i is given by $\text{argmin}_j(D_{ij})$.[3] To impute a new sequence, two key assumptions are made: first, that the sequence's nearest neighbor is one of the observed sequences, and second, that the distance between the sequence and its nearest neighbor is represented in the observed distribution of pairwise distances. Under these assumptions, a new sequence can be generated by drawing a genetic distance from the observed distribution of pairwise distances, then copying an existing sequence and mutating that sequence until it is the desired distance from the original. Throughout this process, it is assumed that the demographic information T_{obs} and F is available; when no demographic information is available, the entire population is considered as one demographic group. In these cases, T_{obs} is a vector assigning every sequence to the category 1, and $F = (1)$. That is, F denotes that the entire population falls into the single available category.

Before imputing, we need to know how many sequences to impute. If no demographic information is available, the number of sequences to impute is simply n_m. If demographic information is available, we can calculate how many sequences should be imputed in each

[3] In the distance matrix as given, each sequence is its own nearest neighbor. To exclude these trivial nearest neighbors, for the rest of the analysis the diagonal elements of D are set to the maximum possible genetic distance $d_{\max}$.

category in order to make the demographic proportions in the data equal to F; denote these numbers as F_{imp}.[4]

The first step in the imputation process is building the distribution of pairwise distances to be drawn from. In particular, we want to sample a plausible distance between the imputed sequence and its nearest neighbor, so the desired distribution is built from the distance between each observed sequence and its nearest neighbor, $N = \{\min_j(D_{ij})|i \in n_o\}$. When demographic traits are given, this additional information can be incorporated by building nearest-neighbor distributions for each pair of demographic groups. $N_{ab} = \{\min_j(D_{ij})|T_i = a, T_j = b\}$ is the distribution of nearest-neighbor distances from a sequence in group a to a sequence in group b. Because N_{ab} and N_{ba} are different distributions (i being j's nearest neighbor does not mean j is i's nearest neighbor), this results in a total of d^2 distributions, though the total number of nearest-neighbor distances is the same regardless of how many distributions these distances are partitioned into.

Next, a specific sequence is imputed. Its demographic category a is drawn from F_{imp}. Next, the demographic category b of its nearest neighbor needs to be chosen. We draw $b \in \{1...d\}$ with probability proportional to $|N_{ab}|$. This guarantees that demographic categories which are most frequently seen as nearest neighbors of sequences in category a are most likely to be chosen as nearest neighbors for the imputed sequence.[5] A nearest-neighbor distance l is then drawn from N_{ab}.

The new sequence s_j is initialized as a copy of its nearest neighbor sequence s_i, chosen uniformly at random from the demographic category b. Mutations are then applied to s_j until $d(s_j, s_i) = l$. In general, these mutations can be applied in very simple ways (such as

[4] If no demographic information is available, sequences will be imputed uniformly at random until there are n_t sequences

[5] If demographic information is not available, only one (trivial) demographic group is considered: the entire population. All imputed sequences are assigned to this demographic category, and their nearest neighbors are drawn uniformly at random from this category, which consists of the entire pool of observed sequences.

making mutations to the sequence uniformly at random), or very sophisticated ones ([27] presents software for simulating sequence evolution along a phylogeny). The method used here strikes a balance between simplicity and sophistication, and has the added benefit that it makes no prior assumptions about mutation frequencies, instead estimating these frequencies from the data. In particular, a $n_t \times |\Sigma|$ matrix W is constructed where each row W_e is a multinomial probability distribution over the characters found at site e in the observed alignment A. Sites are chosen one at a time for mutation; the probability of mutating site e is proportional to $1 - \max(W_e)$. Thus sites that are highly conserved are only rarely mutated while sites that are highly variable are mutated frequently. When a character at a specific site is mutated, the replacement character for that site is drawn as a weighted sample from W_e. This process is repeated until $d(s_j, s_i) = l$.

The new sequence s_j is added to the alignment, and the above steps are repeated to impute new sequences until the alignment has n_t total rows, resulting in an alignment A'_1. Note that as the alignment grows in size, previously imputed sequences may be chosen as nearest-neighbor sequences to be copied and mutated. The process is repeated k times to yield k alignments of size $n_t \times m$. Estimation of alignment properties and variance is performed as described above.

2.3 Evaluation

By definition, it's impossible to test how well a missing-data recovery method performs on a real world dataset because the full data is unavailable. However, given a dataset, the accuracy of a method can be assessed by treating the dataset as complete, deleting a portion of the data, and then attempting to recover the original properties of the dataset. As part

of my research group's work in [24], we evaluated the above method on HIV genetic data collected from Mochudi, Botswana.

The evaluation dataset consists of $n_t = 371$ HIV subtype C sequences collected from the community survey in Mochudi, Botswana mentioned in Section 1.1. The overall percentage of the population which fell into a cluster was used as the estimated property of interest c. Sets of 50, 100, and 200 sequences were repeatedly deleted from the full dataset: for each $n_m \in \{50, 100, 200\}$, 100 different subsampled datasets of $n_o = n_t - n_m$ sequences were generated. In this study, the population was divided into demographic categories – young males, young females, old males, and old females. Females were deleted at a rate 10 times higher than that of males to demonstrate the effectiveness of the method when different population subgroups are missing at different rates. For each subsample, multiple imputation of $k = 10$ datasets was performed 100 times to yield 100 estimates of c as c_{imp} and variances of these estimates, v_{imp}. Thus, for each number of deletions, it is possible to compare 10,000 multiple imputation estimates to estimates resulting from the observed data over a random sample of possible deletions.

For each multiple imputation, we compared the clustering values c calculated from the full dataset to values calculated from only the observed (subsampled) data and to values calculated from a multiple imputation estimate. We also estimated coverage of 95% confidence intervals around the estimates from observed data and from multiple imputation – the proportion of times that a 95% confidence interval around an estimate c_{obs} or c_{imp} contained the true value c. in both cases, the variance of the estimate used to build these intervals was calculated by bootstrapping, and in multiple imputation, the variance was adjusted using the formula from [26] described above. Coverage was calculated as For an unbiased estimator that provided accurate assessments of variance, this proportion would be exactly

95%. In general, the closer the coverage is to 95% the more reliable the estimator. In general, the clustering statistic estimated only from observed data, c_{obs}, is biased downwards. Multiple imputation (c_{imp}) substantially reduces bias (though does not eliminate it), and improves coverage proportions. As more sequences are deleted, estimates from the observed data suffer from an increasing degree of bias. The multiple imputation estimate suffers from less bias, though as missingness becomes severe it does display downward bias as well. The full results, calculated for [24], are presented in Table 2.1 below.

Table 2.1: **Mean clustering statistics c and coverage proportions for full data (c_{true}), observed data (c_{obs}), and multiple imputation estimate (c_{imp}), over 1,000 simulations**

Thresh	c_{true}	# Del	c_{obs}	c_{imp}	Coverage, Observed	Coverage, Imputed
0.1	0.043	50	0.039	0.043	83%	94%
		100	0.036	0.043	66%	89%
		200	0.024	0.037	30%	61%
0.15	0.13	50	0.11	0.13	48%	91%
		100	0.099	0.13	16%	85%
		200	0.059	0.10	1.4%	54%

Thresh indicates the minimum genetic distance (in proportion of total sequence length) below which two sequences were considered part of a cluster. The average clustering value estimated by multiple imputation is closer to the true value than that estimated with the observed data alone, and coverage proportions are also higher.

It is also useful to see the impact of missingness and multiple imputation on estimated statistics through a representative example. Figure 2.1 demonstrates the proportion c of the 371 sequences which are part of any cluster at various genetic distance thresholds. The blue line indicates clustering in the full set of 371 sequences, while the red line indicates clustering for a dataset with 100 sequences deleted. Green points at 0.1 and 0.15 on the horizontal axis represent estimates of c calculated with multiple imputation. Bars indicate 95% confidence intervals. This subset of 271 sequences is representative in the sense that,

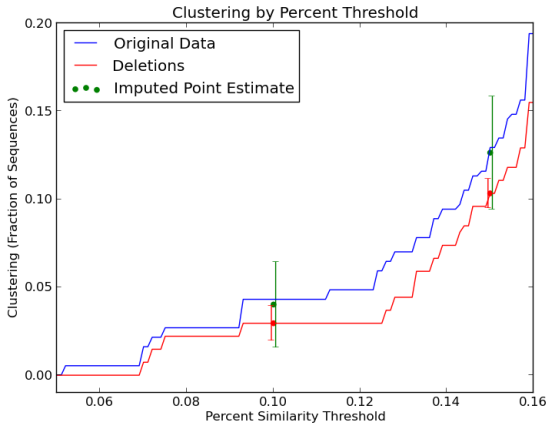

Figure 2.1: A representative deletion of 100 sequences substantially reduces clustering, and imputation recovers much of this bias. Also note that only confidence intervals around the imputed estimates include the true clustering value.

at the 0.1 and 0.15 genetic distance thresholds, the difference between c in the full dataset and the subset is equal to the median difference across 100 different subsamples.

First, calculations using only the dataset that underwent deletions are biased; clustering is substantially lower in the dataset that experienced deletions than in the original data. Second, calculations based on the dataset that experienced deletions underestimate the variance of c; note that the red bars indicating 95% confidence intervals for the value of c using the dataset that experienced deletions do not include the true value. Estimates from multiple imputations are less biased and considerably closer to the true value of c. In addition, these estimates exhibit larger variance and confidence intervals that include the true value of c.

2.4 Discussion

In conclusion, this section has demonstrated that multiple imputation is a useful method for producing estimates of genetic clustering that are less biased and exhibit more accurate assessments of variance. However, it's worth noting that there are limitations to the multiple imputation method. First, while multiple imputation reduces bias, it cannot eliminate it.

One important reason is that the imputation method presented here imputes new sequences in a way that maintains the observed distribution of minimum genetic distances between sequences; however, this observed distribution itself is likely biased by deletions. Because the clusterng proportion is based on the distribution of minimum distances, preserving bias in this distribution will ensure some bias always remains in estimates of c.

The bias in minimum distance distribution can be seen through simulation; in the figure below, subsets of 100 and 200 sequences were repeatedly deleted uniformly at random from the full dataset of 371 sequences. Minimum distances from 100 independent repetitions were pooled and the distributions compared at each number of deletions. Small but statistically significant increases in genetic distance are seen with each increase in the number of deleted sequences. Median genetic distances are 97.0 for the original data (n=371), 98.0 for datasets with 100 deletions (n=2710), and 101.0 for datasets with 200 deletions (n=1710). Differences between distributions were significant as follows:

- Original Data and 100 Deletions: $p < 0.05$ (Mann-Whitney U-test, $U = 472856$)
- Original Data and 200 Deletions: $p < 0.00001$ (Mann-Whitney U-test, $U = 267360$)
- Original 100 Deletions and 200 Deletions: $p < 0.000001$ (Mann-Whitney U-test, $U = 2086828.5$)

(Note that power differs between tests because of different sample sizes).

In addition, while coverage of 95% confidence intervals for imputed estimates of c is a substantial improvement over estimates using only the observed data, imputed coverage still does not reach the ideal of 95 percent. Because coverage of confidence intervals depends both on the estimated value and its variance, this could be representative of bias in imputed

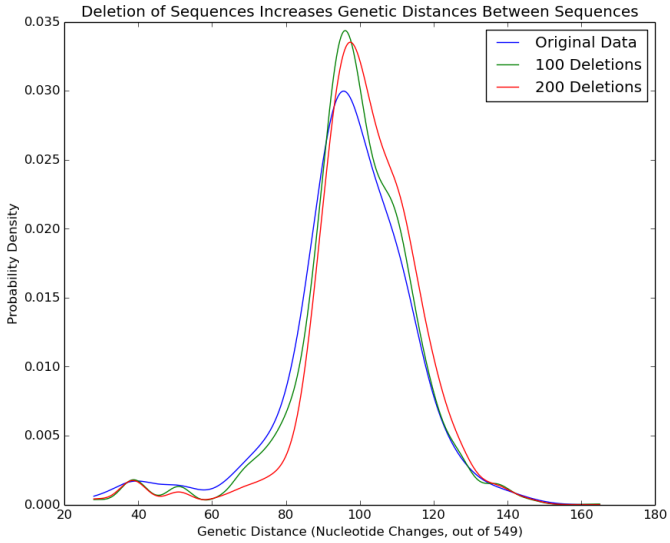

Figure 2.2: Distributions of minimum distances from original dataset of 371 sequences, datasets with 100 sequences deleted, and datasets with 200 sequences deleted. Visual inspection and Mann-Whitney U test confirm that the distribution of genetic distances tends to increase a small but significant amount as more sequences are deleted.

estimates, underestimation of variance, or a combination of the two. Later chapters will focus on estimators with improved bias and variance.

Chapter 3

Subsampling-Based Likelihood Optimization

3.1 Background

This chapter will attempt to address some of the main shortcomings of the multiple imputation framework. One concern about this framework is that it makes multiple assumptions that are difficult to justify. In particular, the random model used to impute new sequences by applying mutations is a drastic oversimplification of real biological evolution. Applying mutations independently at each site with probability proportional to their frequency in the observed data ignores correlations between mutations at different sites and assumes the observed data is a representative sample of mutation frequencies. Another example of a flawed assumption in multiple imputation is the method's reliance on preserving the observed distribution of minimum distances between sequences, though this distribution is biased as discussed at the end of the last chapter. It is of course possible to use many different models

to impute missing sequences, and more realistic models than the one presented in Chapter 2 certainly exist. However, it is unlikely that any model will perfectly replicate sequence evolution, so some biases will persist. Multiple imputation has no way to account for flaws in the imputation model; rather, it blindly averages results over multiple runs of the model. This accounts for the variance of estimates but ignores biases in the model. A second, related concern is that multiple imputation treats all imputed datasets equally by taking an unweighted average over results from each imputation. Some imputed datasets will more accurately fill in missing sequences than others, but the multiple imputation framework has no way to prefer such "higher-quality" datasets.

Both of these concerns could be addressed in a straightforward way if the full dataset A^* were known. Given a statistic of interest c calculated using the function f and an imputed dataset A', we can compare the values of c calculated from the full dataset and imputed dataset, $c^* = f(A^*)$ and $c_{\text{imp}} = f(A')$ respectively. Datasets A' for which $|c_{\text{imp}} - c^*|$ is small are less biased and should be preferred in the estimation process. Though this weighting process is simple, it is also impossible to use in practice because A^* and c^* are unknown. If we knew c^*, we would not have to estimate it at all!

In practice, estimation procedures have access only to the observed, incomplete dataset A and the statistic estimated from this dataset $c_{\text{obs}} = f(A)$. The only information available about the full dataset A^* is that, after undergoing deletion of n_m sequences, it gives rise to the observed alignment A and observed statistic c_{obs}. Intuitively, we should prefer imputed datasets which, when they undergo deletion of n_m sequences, yield observed statistics close to c_{obs}.

3.2 Subsampling Method

Given an imputed dataset A' and full, unobserved dataset A^*, both consisteng of n_t sequences, one way to assess how close A' is to A^* is to subsample n_o sequences from A' a total of k different times, resulting in k incomplete datasets $A^1_{\text{sub}}...A^k_{\text{sub}}$. Calculating the statistic of interest on each of these incomplete datasets yields k estimates of c, $c^1_{\text{sub}}...c^k_{\text{sub}}$. The average of these estimates, or

$$c_{\text{sub}} = \frac{\sum_{i=1}^{k} c^i_{\text{sub}}}{k}$$

approximates the expected value of c when estimated from any subsample of n_o sequences from A'. Datasets A' for which $|c_{\text{sub}} - c_{\text{obs}}|$ is small are favorable because they are similar to A^* in the one way we can observe: when they undergo deletions, they result in similar estimates of the statistic c. See Figure 3.1 for an illustration of this comparison process.

While it can be shown that imputed datasets that minimize $|c_{\text{sub}} - c_{\text{obs}}|$ produce more accurate estimates of c, the resampling procedure used to calculate c_{sub} poses several problems. First, repeated subsampling and calculation of c for each subsample can be computationally expensive. Second, it is unclear *a priori* how many subsamples are required to produce a stable estimate of the expected subsampled value of c, or c_{sub}. Fortunately, for the statistic we are interested in calculating (the proportion of sequences which are part of a cluster when clusters are defined using pairwise genetic distance between sequences), it is possible to exactly calculate c_{sub} in closed form without subsampling.

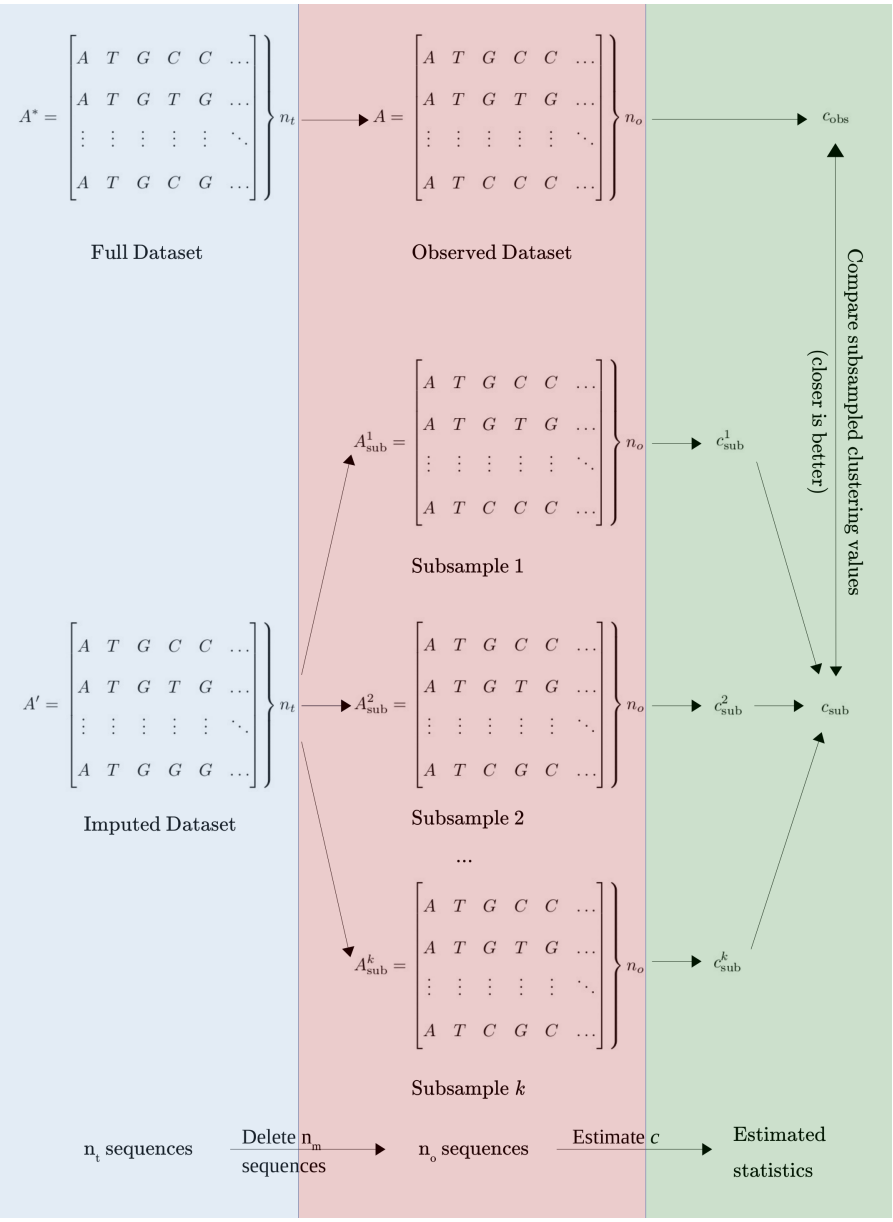

Figure 3.1: Diagram of a process for assessing quality of an imputed dataset. Both the imputed dataset and the complete data have the same number of sequences, but the complete dataset is unavailable for comparison. The imputed dataset is subsampled multiple times, resulting in many datasets of the same size as the observed data. The average estimate of c, or c_{sub} across these datasets is compared to the observed value of c, or c_{obs}. The more similar these values are, the more similarly the imputed dataset behaves to the full dataset when subjected to deletions and the more likely it is to produce a correct estimate of c.

3.3 Closed-Form Calculation

Consider an imputed $n_t \times m$ alignment A'. Recall that the pairwise distance matrix, here denoted D', is constructed by measuring the genetic distance between all pairs of sequences; for distance metric d, $D'_{ij} = d(s_i, s_j)$. Let $d_{\max}$ be the maximum possible genetic distance; because we generally want to ignore sequences whose only neighbor is themselves, it is generally convenient to set $d(s_i, s_i) = d_{\max}$. For some threshold $t \in (0, 1)$, any two sequences s_i and s_j for which $D'_{ij} \leq td_{\max}$ are considered "linked" and will be part of the same cluster. The statistic of interest c is the proportion of sequences which are part of any cluster; that is, which are close enough to any other sequence to share a cluster with it. The clustering matrix M' is a binary matrix with entries defined as 1 if the distance between two sequences falls below the threshold and 0 otherwise: $M'_{ij} = I(D_{ij} \leq td_{\max})I(i \neq j)$. The statistic c is then simply the proportion of rows that contain a 1 at any point, or

$$c = \frac{\sum_{i=1}^{n_t} I(\sum_{j=1}^{n_t} M_{ij} \geq 1)}{n_t}$$

This is simple to calculate with a single pass over the matrix M'. However, we want to calculate not just c but c_{sub}, which is the expected value of c calculated on a subsample of n_o sequences from A'. In other words, if c were calculated using each of the $\binom{n_t}{n_o}$ possible subsampled alignments, its mean value would be c_{sub}. It is possible to approximate c_{sub} by randomly generating subsamples of A', but exactly calculating c_{sub} this way is wildly impractical. For the 371-sequence sample considered here, a typical subsample where 100 sequences are missing yields $\binom{371}{271}$ possible subsamples, or roughly 3.73×10^{92}.

Because the process of calculating c involves considering interactions between all pairs of sequences, it's not immediately clear how to calculate c_{sub} without individually examining

each subsample. However, by using linearity of expectation c_{sub} can indeed be calculated in closed form. Let A be a matrix-valued random variable which is a $n_o \times m$ matrix drawn uniformly at random from all subsamples of n_o sequences from A'. Denote the corresponding pairwise distance matrix as D and the clustering matrix as M. Then:

$$c_{\text{sub}} = \mathbb{E}\left(\frac{\sum_{i=1}^{n_o} I(\sum_{j=1}^{n_o} M_{ij} \geq 1)}{n_o}\right)$$

By linearity, we can move the expectation inside the sum:

$$c_{\text{sub}} = \frac{\sum_{i=1}^{n_o} \mathbb{E}\left(I(\sum_{j=1}^{n_o} M_{ij} \geq 1)\right)}{n_o} = \frac{\sum_{i=1}^{n_o} P(\sum_{j=1}^{n_o} M_{ij} \geq 1)}{n_o} \tag{3.1}$$

This expression still depends on the specific form of the clustering matrix M corresponding to the subsample A, however. We would like it to depend only on M', the full clustering matrix. We focus on the term $P\left(\sum_{j=1}^{n_o} M_{ij} \geq 1\right)$:

$$P\left(\sum_{j=1}^{n_o} M_{ij} \geq 1\right) = \sum_{i'=1}^{n_t} P\left(\sum_{j=1}^{n_o} M_{ij} \geq 1 | A_i = A'_{i'}\right) P\left(A_i = A'_{i'}\right)$$

It is hard to calculate the probability that an arbitrary row of a subsampled clustering matrix will sum to more than one, but it is relatively easy to calculate this probability when we know which sequence the row corresponds to; that is, when we condition on a given sequence from the original alignment. Given a specific sequence, this probability is equal to the probability that at least one neighbor of that sequence is also sampled; that is, the

complement of the probability that no neighbors are sampled:

$$P\left(\sum_{j=1}^{n_o} M_{ij} \geq 1 | A_i = A'_{i'}\right) = 1 - P\left(\sum_{j=1}^{n_o} M_{ij} = 0 | A_i = A'_{i'}\right) = 1 - \left(\frac{n_m}{n_t}\right)^{\sum_{j=1}^{n_t} M'_{i'j}} \tag{3.2}$$

Because sampling happens uniformly at random, each sequence is chosen to be part of the subsample independently of the other sequences. Thus the probability that no neighbors of sequence i' are sampled is the probability of any individual sequence not being sampled $\left(\frac{n_m}{n_t}\right)$ raised to the power of the number of neighbors sequence i' has. The numerator on the RHS of (3.1) can now be written as

$$\sum_{i=1}^{n_o} P\left(\sum_{j=1}^{n_o} M_{ij} \geq 1\right) = \sum_{i=1}^{n_o}\sum_{i'=1}^{n_t}\left(1 - \left(\frac{n_m}{n_t}\right)^{\sum_{j=1}^{n_t} M'_{i'j}}\right) P(A_i = A_{i'})$$

All sequences in the alignment A' are equally likely to end up as a given sequence A_i in A, so $P(A_i = A_{i'}) = \frac{1}{n_t}$. Upon making this substitution, the expression no longer depends on i, so we can simplify:

$$\sum_{i=1}^{n_o}\sum_{i'=1}^{n_t}\left(1 - \left(\frac{n_m}{n_t}\right)^{\sum_{j=1}^{n_t} M'_{i'j}}\right) P(A_i = A_{i'}) = \sum_{i'=1}^{n_t}\frac{n_o}{n_t}\left(1 - \left(\frac{n_m}{n_t}\right)^{\sum_{j=1}^{n_t} M'_{i'j}}\right)$$

The full equation for c_{sub} from the RHS of (3.1) can now be written as:

$$c_{\text{sub}} = \frac{\sum_{i=1}^{n_t}\frac{n_o}{n_t}\left(1 - \left(\frac{n_m}{n_t}\right)^{\sum_{j=1}^{n_t} M'_{ij}}\right)}{n_o} = \sum_{i=1}^{n_t}\frac{1}{n_t}\left(1 - \left(\frac{n_m}{n_t}\right)^{\sum_{j=1}^{n_t} M'_{ij}}\right) \tag{3.3}$$

This expression involves just two sums, which can be calculated by passing over the clustering matrix M' for the full alignment A' a single time. This is in stark contrast to the

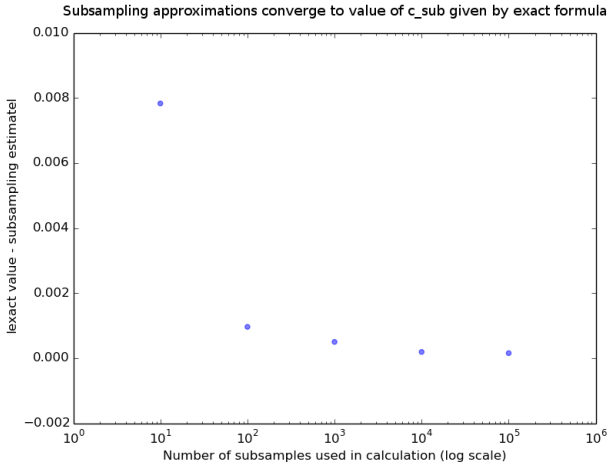

Figure 3.2: Convergence of subsampled c_{sub} estimates to exactly calculated value. X-axis represents number of subsamples used to estimate c_{sub} (estimated subsampled clustering value using Hamming distance, as used throughout this paper). Y-axis represents absolute difference between subsampling estimate and exactly calculated value for c_{sub}. As the number of subsamples used increases, the estimate of c_{sub} converges to the exactly calculated value. Dataset is 200 HIV-C *env* gp120 V1C5 sequences collected from Mochudi, Botswana.

combinatorial explosion of terms involved in calculating c_{sub} by subsampling. Figure 3.2 below demonstrates that values of c_{sub} calculated using subsampling converge to the value calculated with the exact formula. The key property that allows for closed-form calculation of c_{sub} is that $d(s_i, s_j)$ is independent of other sequences, that is, of all s_k for which $k \neq i, k \neq j$. This property is what lets us relate M_{ij} and $M_{i'j}$ in equation (3.2); the probability that sequence i in A has at least one neighbor depends entirely on whether one of its neighbors in the original alignment A' was sampled. It's worth noting that not all distance metrics have the property that $d(s_i, s_j)$ depends only on s_i and s_j; in particular, for distances calculated using a phylogeny, $d(s_i, s_j)$ depends on all other sequences in the alignment as well. However, many distance metrics, including those based on Hamming distance as well as popular nucleotide and amino acid substitution models, respect this independence assumption and allow closed-form calculation of c_{sub}.

3.4 Evaluation

3.4.1 Data

Because the work in this chapter and those succeeding it occurred at a later date than the work presented in Chapter 2, a new dataset was available for analysis. All subsequent results are based on a dataset consisting of a 1248-sequence, 1797-nucleotide alignment of the V1C5 region of HIV-C *env* gp120 sequences collected from patients in Mochudi, Botswana. Because the methods presented later in this thesis are, in general, more computationally intensive, subsets of the full dataset are often chosen. As there are many possible subsets, many (generally 100) subsamples are considered and the subsample with the median clustering proportion c is chosen.

3.4.2 Results

As can be seen in Figure 3.3, minimizing $|c_{\text{sub}} - c_{\text{obs}}|$ does indeed reduce absolute error in estimating c over 100 different 30-sequence subsamples of a 50-sequence dataset (all sequences from the dataset discussed immediately above). The median correlation between $|c_{\text{sub}} - c_{\text{obs}}|$ and absolute error in estimating c is $r = 0.38$. The heterogeneity of behavior under subsampling is surprising, however. A substantial number of subsampled datasets exhibit behavior we would expect of rare, pathological cases: for these subsamples, $|c_{\text{sub}} - c_{\text{obs}}|$ actually demonstrates a strong negative correlation with error, implying that datasets which tend to subsample in the same way we observed in the original dataset actually do worse at estimating c. Interestingly, subsamples that have correlation coefficients close to zero can be easily separated into subsets of points within which strong correlations exist. Some subsamples exhibit a line of positively correlated points and a line of negatively correlated

points; others have two offset positively correlated lines and a single set of negatively correlated points. This implies that, while the behavior of sequence datasets when subsampled is complex, this behavior can be decomposed into combinations of a few simple behaviors. Future studies into the structure of these correlations could yield valuable information. In addition, that the correlation is somewhat weaker than expected supports the expansion of this method by comparing not just one subsampled statistic but many. If weak positive correlations hold for all of these statistics, favoring datasets that minimize $|c_{\text{sub}} - c_{\text{obs}}|$ for many statistics c should minimize error even further in estimating all properties of interest.

These results also have implications for how we should use information about c_{sub} to inform missing data analysis. Given that there is a correlation between $|c_{\text{sub}} - c_{\text{obs}}|$ and estimation error, one approach to dealing with missing data would be to impute a dataset and iteratively optimize it to minimize $|c_{\text{sub}} - c_{\text{obs}}|$. This is the best dataset to analyze in the sense that it behaves most similarly to the original data in the ways that we can observe (that is, its behavior under subsampling). Though the results in this section imply that datasets generated using such a method would be more likely to be accurate than an arbitrarily chosen imputed dataset, the wide range of correlations observed in Figure 3.3 imply that it will still often yield inaccurate estimates, because in a substantial minority of subsamples $|c_{\text{sub}} - c_{\text{obs}}|$ exhibits a negative correlation with error. In these cases, the optimized dataset is being optimized in the wrong direction! While a single dataset which minimizes $|c_{\text{sub}} - c_{\text{obs}}|$ is an improvement over a random imputed dataset, we can still do better. Generating a *distribution* of datasets, each optimized to different values of $|c_{\text{sub}} - c_{\text{obs}}|$ and weighted accordingly, will give a more accurate assessment of the possible variation in our estimates.

Figure 3.3: Minimizing $|c_{\text{sub}} - c_{\text{obs}}|$ favors datasets that minimize error in estimating c. This set of simulations started with a dataset of 50 sequences $\times$ 1797 nucleotides; out of 100 such subsamples of the full dataset, the one with median clustering value c was chosen for representativeness. One hundred different subsamples of 30 sequences were taken, and 100 imputations of 20 new sequences performed on each. For each subsampled dataset, a scatter plot is shown of the absolute difference between the subsampled clustering value c_{sub} for the imputed dataset and the clustering value observed in the subsampled dataset of 30 sequences c_{obs} (x-axis) and the absolute error in the imputed dataset's estimate of c (plots sorted by decreasing correlation coefficient). Most datasets demonstrate a positive correlation between $|c_{\text{sub}} - c_{\text{obs}}|$ and error in the estimate of c; for some datasets this correlation is smaller and in some cases it is negative. These exceptions are expected in our model; some subsets of the data are outliers in terms of their values of c_{obs} and are not representative of the typical behavior of the orignal dataset under subsampling. Overall, the relationship between $|c_{\text{sub}} - c_{\text{obs}}|$ and estimation error is usually positive, with mean $r = 0.17$ and median $r = 0.38$. Finally, in two cases, subsampling reduces clustering to zero and imputation is incapable of recovering any clustering at all. In these cases (the last two in the figure), correlation is undefined. These cases are excluded from analysis in calculating the overall value of r.

Chapter 4

Distribution-Based Approaches

The resampling-based approach presented in the last section demonstrates reduced bias and error in recovering clustering estimates, in that imputed alignments A' that minimize $|c_{\text{sub}} - c_{\text{obs}}|$ are more likely to produce accurate estimates of c. However, it is not immediately clear how to incorporate this knowledge into a method that not only provides estimates of c but also allows for construction of accurate confidence intervals around the estimate. The approaches presented below seek to recover a distribution over the true value of c that incorporates as much information as possible from the observed data.

4.1 A Full Distribution over Missing Data

It is relatively clear that, because datasets with lower values of $|c_{\text{sub}} - c_{\text{obs}}|$ tend to produce more accurate estimates, they should be weighted more highly. However, it is difficult to justify any particular weighting scheme over any other. For example, $w = \frac{1}{|c_{\text{sub}} - c_{\text{obs}}|}$ (used above) and $w = 1 - |c_{\text{sub}} - c_{\text{obs}}|$ both prefer datasets with smaller differences between the subsampled and observed statistics of interest, but it is unclear which should be preferred.

The arbitrary nature of the weighting function underlies a fundamental question: can we generate a distribution over all possible datasets that fill in missing data representing our belief that a given dataset represents the "truth"?

The problem is not at all trivial; first, it requires exploring a massive space. In the examples in Chapter 2, the full dataset consists of 371 sequences and a deletion of 100 is considered, so the goal is to use 271 observed sequences to impute 100 more. Considered as nucleotides, these 100 sequences consist of 1647 bases; each can be one of five characters (A,C,T,G, or gap). The total number of possible ways to fill in these sequences is the number of ways to generate one hundred 1647-character strings from an alphabet of 5 characters, or $5^{100\times1647} \approx 10^{427096}$ (for comparison, there are only 10^{81} atoms in the observable universe). Second, the problem requires constructing a reasonable way to weight each of these possible datasets so that it is possible to make quantitative measurements of how much more likely one dataset is than another. Finally, a method for drawing datasets according to this metric from a huge sample space is required to obtain actual numerical results.

The overall process can be split into two steps: formulating a desired distribution over all possible datasets, and approximating this distribution by sampling. As in Chapter 3, our only source of information about the full distribution is the subsample we observe and the statistic estimated from it, c_{obs}. Thus, rather than specifying the relative frequencies with which each dataset should be drawn, we can specify the relative frequencies with which datasets with given values of c_{sub} should be drawn. The idea that datasets minimizing $|c_{\text{sub}} - c_{\text{obs}}|$ result in the most accurate estimates implies our distribution should place the most weight on these datasets. In addition, by bootstrapping the columns of the observed alignment A, we can estimate the variance of the statistic c_{obs} and require that the target distribution share this variance. This information is enough to specify a Normal distribu-

tion over c_{sub} that determines what proportion of the desired sample has various expected clustering values after subsampling. We will call this *proxy distribution* P; the goal is to draw datasets so that the values of c_{sub} corresponding to each dataset fit the distribution P. This collection of datasets represents our belief about the likely ways to fill in the missing sequence data. Given such a collection of datasets, it is possible to calculate an estimate of c, by averaging over $c_i = f(A'_i)$ for each dataset A'_i in the distribution. Similarly, the variance of the estimate is the variance of the c_i. Higher order moments can also be estimated because the datasets A'_i are samples from the distribution over all possible datasets.

The challenge that remains is to draw samples from this distribution. The following sections present two sampling methods for this task. The first, a Markov Chain Monte Carlo based approach presented in Section 4.2, was ultimately unsuccessful. The second, based on optimizing a distribution of datasets for similarity to the distribution P, was successfully implemented and is presented in Section 4.3.

4.2 MCMC Sampling

4.2.1 Overview of MCMC

Our goal is to draw representative samples from a massive, high-dimensional space. A natural method for this purpose is Markov Chain Monte Carlo (MCMC): given a way to step from state to state in the space and a way to evaluate the relative probabilities of any two states up to a proportionality constant, MCMC can draw samples approximately from and asymptotically converging to an arbitrary probability distribution. The specific algorithm considered in this section is Metropolis-Hastings, which represents one way to construct a Markov chain with the desired stationary distribution [28].

The first step in implementing Metropolis-Hastings is to initialize the first state of the Markov chain. In this case, the states being considered are datasets, so the first state will ideally be a candidate full dataset of relatively high probability. The multiple imputation method discussed in Chapter 2 provides a convenient method for generating a plausible initial state; simply impute a single $n_t \times m$ dataset A'_1 as the first step of the chain.

To generate the rest of the chain, we require a rule for stepping from state to state which will cause the chain to converge to the target distribution T. Given a state A'_i, the key idea of Metropolis-Hastings is to propose a new state A'_{prop}. The next state in the chain state A'_{i+1} will be chosen with some probability from A'_i and A'_{prop}, and the process will begin again. The *acceptance probability* a is the probability that A'_{prop} will become the next state A'_{i+1}; if not, we set $A'_{i+1} = A'_i$ and continue. The value of a is defined as

$$a = \min\left(\frac{T(A'_{\text{prop}})}{T(A'_i)}\frac{Q(A'_{\text{prop}} \to A'_i)}{Q(A'_i \to A'_{\text{prop}})}, 1\right) \tag{4.1}$$

where $T(A'_i)$ and $T(A'_{\text{prop}})$ represent the likelihoods of the current state and proposed new state in the target distribution and only need be determined up to a constant of proportionality. That is, $T(A'_i)$ and $T(A'_{\text{prop}})$ do not need to be exactly determined so long as the ratio $\frac{T(A'_{\text{prop}})}{T(A'_i)}$ is correct. $Q(A'_i \to A'_{\text{prop}})$ represents the probability of proposing state A_{prop} from state $T(A'_i)$ and vice versa. Taking the minimum with 1 guarantees that a does not exceed 1 and remains a valid probability. Hastings showed that for an arbitrary target probability distribution and proposal method, chains whose steps satisfy this acceptance probability for stepping to a new state will converge to the target distribution so long as they are ergodic (aperiodic and with finite expected return time for each state) [28]. The next two sections will discuss methods for proposing new states and determining what target distribution the chain should converge to.

4.2.2 Proposal Methods

Given some state A'_i which is a full dataset of n_t sequences that fills in the missing data, there are many ways to propose a new state A'_{prop}. The only absolute requirements are that the proposal method alters only the n_m missing sequences, not the n_o observed sequences (as these are known to represent sequences actually present in the population), and that the likelihood of steps between states can be calculated up to a proportionality constant (so that it is possible to calculate the term $\frac{Q(A'_{\text{prop}} \to A'_i)}{Q(A'_i \to A'_{\text{prop}})}$). Three proposal methods are easily available based on previous work and the techniques presented in this thesis:

- **Uniform Proposal:** Simply make a certain number of random mutations for each proposal. Pick k sites to mutate, uniformly distributed across the n_m unobserved sequences and m sites in each sequence, and mutate the character currently present at each site to another character uniformly at random.

- **Transition Matrix:** Pick k sites to mutate as in the uniform proposal. However, instead of mutating the characters at these sites uniformly at random, use a transition matrix H that allows for differential rates of mutation between different characters, so that H_{ij} represents the probability of mutation from character i to character j. Many such transition matrices have been constructed from empirical data for phylogenetic analysis, and can readily be applied in this proposal method. This approach has the advantage of making mutations that are more frequently observed in real sequences.

One important advantage of these first two methods is that they can be used to generate *symmetric* proposals; that is, proposals in which $Q(A'_{\text{prop}} \to A'_i) = Q(A'_i \to A'_{\text{prop}})$. Such methods are advantageous because there is no need to explicitly calculate $\frac{Q(A'_{\text{prop}} \to A'_i)}{Q(A'_i \to A'_{\text{prop}})}$; it will always be equal to one. Because both of the above methods involve making independent

mutations, the likelihood of a given proposal is simply the product of the probabilities of all mutations that are made. In the uniform proposal, all mutations are equally likely, so the likelihood of mutating A'_i into A'_{prop} is exactly the same as the likelihood of mutating A'_{prop} back into A'_i. The transition matrix proposal method is also symmetric if a symmetric transition matrix is used, as the probability of each individual mutation H_{ij} made in the proposal $Q(A'_i \to A'_{\text{prop}})$ is equal to the probability of a mutation H_{ji} made in the reverse step $Q(A'_{\text{prop}} \to A'_i)$. It happens that most transition matrix models used in phylogeny are symmetric, which is convenient for this application as well.

Of course, the symmetry assumption may not be very biologically realistic. One way to relax this assumption is to use a non-symmetric transition matrix. While we can no longer avoid calculating $\frac{Q(A'_{\text{prop}} \to A'_i)}{Q(A'_i \to A'_{\text{prop}})}$ in this case, the procedure is relatively simple: to calculate $Q(A'_i \to A'_{\text{prop}})$, simply multiply the probabilities H_{ij} for each mutation made, and do the same with the probabilities of reverse mutations H_{ji} to calculate $Q(A'_{\text{prop}} \to A'_i)$. One more proposal method is easily available:

- **Position-Specific Mutations:** In fact, we already developed a system for mutating sequences in Chapter 2. Recall that we generated a $n_t \times |\Sigma|$ matrix W whose rows are empirically derived probability distributions over the characters at each site in the observed alignment A. We can choose sequences uniformly to mutate; then, as in the multiple imputation method, we can choose to mutate site e with probability proportional to $1 - \max(W_e)$ and replace the character at site e by drawing from W_e.

This method is not a symmetric proposal; however, because mutations are independent, the likelihood $Q(A'_i \to A'_{\text{prop}})$ of any given set of mutations is simply the product of the individual mutation likelihoods $\prod_e (1 - \max(W_e)) W_{ed'}$ where d' is the index of the character chosen to replace e. The reverse likelihood $Q(A'_i \to A'_{\text{prop}})$ is the product of the likelihoods

of mutating all sites back to their original characters: $\prod_e (1 - \max(W_e))W_{ed}$ where d is the original character found at site e.

4.2.3 Target Distribution

The biggest challenge in using MCMC for sampling in this situation is specifying the distribution over datasets to be targeted, or T in equation (4.1). It is important to distinguish that the proxy distribution P presented in Section 4.2.1 is a distribution over *subsampled clustering values* c_{sub}, not over datasets. It is not trivial to turn a distribution over the desired subsampled statistics c_{sub} into a distribution over datasets A'. In order to generate datasets whose c_{sub} statistics match the distribution P, we do not need to place any relative preference on different proposed datasets that have the same value of c_{sub}. However, determining the relative weights to place on proposed datasets that have *different* values of c_{sub} requires knowing how many possible full datasets A' have each value of c_{sub}. Sampling according to a distribution such as P, which considers only the desired values of c_{sub} and not the number of datasets corresponding to each such value, will overrepresent values of c_{sub} that correspond to many possible datasets, and underrepresent values of c_{sub} that correspond to relatively few datasets.

A simple example can help to illustrate this problem. Consider two c_{sub} values c^1_{sub} and c^2_{sub} from the desired distribution P over c_{sub} values, and suppose P assigns the respective likelihoods $P(c^1_{\text{sub}}) = p$ and $P(c^2_{\text{sub}}) = 2p$. Using P alone to assign weights will result in datasets with subsampled c-value c^2_{sub} being selected at twice the rate of datasets with value c^1_{sub}. However, suppose that there are four times as many datasets with value c^1_{sub} as with value c^2_{sub}. Even though these datasets are half as likely to be selected in any given comparison as datasets with value c^2_{sub}, because they are four times as common they will

actually be represented twice as often in the final sample! Thus, a correction is needed to preserve the desired frequencies of given c_{sub} values.

An effective correction is to define the value of the target distribution at a given value of c_{sub} as $T(c_{\text{sub}}) = \frac{P(c_{\text{sub}})}{|c_{\text{sub}}|}$, where $|c_{\text{sub}}|$ denotes the number of datasets with subsampled statistics c_{sub}. Unfortunately, $|c_{\text{sub}}|$ is nearly impossible to calculate. Several methods have thus far been ineffective:

- Random sampling of possible datasets to estimate the $|c_{\text{sub}}|$ sizes is impractical. The space of all possible datasets, or of all $n_t \times m$ matrices with $|\Sigma|$ possible characters in each entry, is nearly infinite. In this study the statistic of interest c is based on genetic distance. However, the Hamming distance between long sequences of characters in datasets generated uniformly at random will converge rapidly to $\frac{1}{|\Sigma|}$. Thus, if sequences are sufficiently long, almost all sampled c_{sub} values will be very close to 0 or 1, depending on whether the threshold genetic distance is greater or less than $1 - \frac{1}{|\Sigma|}$. Usually, these random datasets will have pairwise distances far above the threshold and no sequences will be part of any cluster; this will make it nearly impossible to sample datasets with other subsampled statistics.

- Given that the vast majority of possible datasets are likely to have 0 clustering, it seems possible to assume $|c_{\text{sub}}|$ exponentially decays from some starting value at $c_{\text{sub}} = 0$ and, through heuristics and approximations, attempt to guess the exact form of this decay. So far, no such guess has been effective in practice.

- It is relatively simple to calculate how many datasets of size $n \times m$ have a given clustering value c under the Hamming distance metric: the number of possible sequence pairs is $n_c = \binom{n}{2}$, while the probability of any individual pair clustering can be calculated

using the Binomial distribution: $p_c = (1 - B((1-t)m))$ where $B = \text{Binom}(m, \frac{1}{|\Sigma|})$. A distribution over the number of sequences that cluster can then be written exactly as $\text{Binom}(n_c, p_c)$ or approximately as $\text{Poisson}(n_c p_c)$. However, while it is easy to calculate the distribution of values of c in this way, c_{sub} is more challenging to calculate because it involves subsampling of the dataset. Many different types of datasets can have the same clustering value c, datasets with different values of c can subsample down to the same value of c_{sub}, and datasets with the same value of c can subsample down to different values of c_{sub}. For these reasons, we have been unable thus far to analytically express or even approximate a distribution over c_{sub}.

Because we have been unable to find a way to calculate $|c_{\text{sub}}|$, it is impossible to construct a target distribution that will sample datasets in a way which yields the desired distribution P of c_{sub} values. The next section will present an approach which more successfully draws samples from this distribution. It is worth noting that it certainly seems possible to calculate the distribution of $|c_{\text{sub}}|$ over all datasets, though we have been unsuccessful thus far. In particular, moving from the known distribution over the number of datasets with a given value of c to a distribution over the number of datasets with a given value of c_{sub} still seems promising. However, it is also worth noting that even if this distribution can be analytically determined, the generality of the MCMC approach is limited. Even for the simple statistics c and c_{sub} defined by Hamming distance between sequences, it is hard to calculate the relative sizes of $|c_{\text{sub}}|$; for arbitrary properties c there is no guarantee that finding a distribution over c_{sub} will be possible. Thus, while the MCMC approach has promising statistical foundations and useful properties, its reliance on knowing the sizes of $|c_{\text{sub}}|$ makes it less valuable than a more broadly applicable method. An approach that does not rely on calculating $|c_{\text{sub}}|$ will be presented in the next section.

4.3 Sampling by Distribution Optimization

Our new goal is to obtain a collection of datasets with some desired distribution over the subsampled statistics c_{sub} which does not require knowledge of how many datasets correspond to each value of c_{sub}. A more tractable way to frame this problem is to note that, instead of considering our states as individual datasets and altering them in a way which prefers more likely datasets as in MCMC, we can consider our states to be *distributions* of datasets and iteratively improve a distribution to match the one we desire.

Formally, we initialize a collection of k datasets $\mathbf{A'_1} = \{A'_1 ... A'_k\}$, each an $n_t \times m$ matrix. By calculating c_{sub} for each matrix we obtain a distribution of subsampled statistics $\mathbf{C^1_{\text{sub}}} = \{c^1_{\text{sub}} ... c^k_{\text{sub}}\}$. The goal is to iteratively optimize the distribution of datasets so that at step i and dataset distribution $\mathbf{A'_i}$, the corresponding distribution of subsampled statistics $\mathbf{C^i_{\text{sub}}}$ closely matches the desired distribution of subsampled statistics P (see Section 4.2). In particular, because we seek to maximize the similarity between the distributions $\mathbf{C^i_{\text{sub}}}$ and P, we can frame the problem as minimizing the Kolmogorov-Smirnov (KS) distance between the two distributions. The KS-statistic is defined as the maximum difference between CDFs of two distributions and is frequently used as a non-parametric test for similarity of distributions [29]. A KS distance of zero corresponds to identical distributions; if the CDFs never differ by more than zero, the distributions must be the same.

For initial evaluation of this proposed method, we present a simple hill-climbing algorithm for optimizing the two distributions. The initial collection of datasets $\mathbf{A'_1}$ can be created using the multiple imputation procedure from Chapter 2. At step i, we have a collection of datasets $\mathbf{A'_i}$. A new collection of datasets $\mathbf{A'_{\text{prop}}}$ is created by using one of the three proposal methods from Section 4.3.2 (uniform, transition matrix, or position-specific proposals). The KS distances $KS(\mathbf{C^i_{\text{sub}}}, P)$ and $KS(\mathbf{C^{\text{prop}}_{\text{sub}}}, P)$ are compared; if the KS

distance for the proposed dataset is smaller, we set $\mathbf{A'_{i+1}} = \mathbf{A'_{prop}}$. Otherwise, we stay in the same state and $\mathbf{A'_{i+1}} = \mathbf{A'_{i}}$. The process repeats until either $KS(\mathbf{C^{i}_{sub}}, P) = 0$, until convergence (no further improvements in KS-value are made), or until an upper limit in the number of steps has been reached. Our implementation used a transition-matrix based proposal method and optimized a distribution of 100 total imputed datasets. The optimization procedure ran for 1000 steps or until convergence.

Table 4.1: **Mean Squared Error and Coverage of Optimization and Imputation Methods**

MSE (Observed)	MSE (Imp)	MSE(Opt)	Coverage (Imp)	Coverage (Opt)
0.0116	0.00805	0.00679	99.9%	88.2%

Mean squared errors and coverage percentages are presented for imputation and optimization methods. Datasets of 30×100 sequences & nucleotides were drawn uniformly at random 100 times and 10 different subsamples of 20 sequences were taken from each dataset, so that the procedure tested variation over initial datasets as well as over subsamples. Clustering values c of the datasets ($t = 0.12$) ranged from 0.1 to 1.0, while clustering values c_{obs} in the subsamples ranged from 0.0 to 1.0. Mean and 95% confidence intervals were constructed for imputation as in Chapter 2, and for optimization from the mean and percentiles of the simulated distribution. Optimization of a distribution of 100 imputed datasets ran for 1,000 time steps. The ability of imputation and optimization methods to recover c was compared to estimates from the obsered data alone. Imputation improved mean squared error over estimates from the observed data, and optimization improved mean squared error over imputation by roughly the same amount. Also of note is that imputation overestimated the width of confidence intervals while optimization underestimated them.

Results of evaluation of the optimization method over 100 random datasets and 10 subsamples per dataset demonstrates reduced mean-squared error. As expected, imputation reduces mean squared error over using the observed data alone, and optimization reduces mean squared error relative to imputation. Variances to construct confidence intervals were built by bootstrap for imputation, as in Chapter 2. For the optimization method, confidence intervals were built empirically from the 5th and 95th percentile clustering values of the generated distribution.

Optimization tended to underestimate the size of confidence intervals (mean size ± 0.20); this is not unexpected, as the hill-climbing scheme likely converged to a local minimum too soon to account for full variance of the target distribution. Until convergence to the target is reached, the optimized distribution is still close to an imputed distribution, for which confidence intervals constructed solely from the variance in estimated values of c will not include within-imputation variance and will be too small (see Section 2.1, [26]).

More surprisingly, confidence intervals constructed around imputed estimates appear to be too large (mean size ± 0.25), despite the fact that in Chapter 2 multiple imputation underestimated confidence intervals when substantial missingness occurred. This phenomenon, combined with the fact that mean squared errors are low across estimates from observed data, imputation, and optimization, may suggest that the datasets are too small to draw strong conclusions about the effectiveness of these methods. The bias induced by deletion may be more visible in larger datasets, and the variance induced by sequence deletion may be less severe (both of these trends are illustrated in Section 1.3).

While this method demonstrates reduced mean-squared error, improving over the multiple imputation approach presented in Chapter 2, it is worth noting that this optimization method is very simple and serves mainly to illustrate the promise of this method. In long runs of the optimization function, values tended to stabilize after roughly 1,000 steps; however, hill-climbing in this high-dimensional space is almost certain to be confounded by local minima. Future work should explore more sophisticated global optimization methods that can avoid local minima. The multiple-sequence-alignment framework is particularly well-suited to approaches like genetic algorithms that mutate and recombine solutions.

Chapter 5

Discussion and Conclusions

This thesis has presented several methods for imputing missing data in genetic sequence datasets and assessing the quality of the resulting estimates. This section will briefly discuss the strengths and weaknesses of the methods in the previous chapters and suggest directions for future research.

5.1 Broader applications

Though these methods were developed for dealing with HIV sequence data and have shown success in this application area, they are generally applicable to any datasets with missing values. Multiple imputation has been a standard approach to missing data for a long time; however, the subsampling- and optimization-based procedure presented here is novel and presents a general framework into which many different models for data and missingness can be incorporated. In general, applying the optimization method to an arbitrary dataset involves three steps:

- Initialize a distribution of full datasets from the observed data. The model used to generate these datasets can be as sophisticated as desired, and in general using domain knowledge to generate a more plausible distribution will make the process more effective. However, even a simple mechanism like adding noise to existing data (which is essentially the implementation used in Chapter 2) should be effective as the iteration will ideally lead to a similar optimum.

- Implement a model of missingness to be used in the subsampling procedure. This step is essential because if missingness is known to be nonrandom but no information about the mechanism is available, subsampling will favor datasets that behave similarly to the true dataset under a *different* subsampling process. This will not result in improved estimation accuracy. However, if information about the missingness mechanism is available, this method can address MAR and NMAR missingness. Even missingness that depends on unobserved characteristics of the data can be educatedly corrected for, as the score of a dataset is based on how the full dataset, including the imputed missing data, subsamples.

- Iteratively modify the distribution of full datasets, seeking an optimum in terms of the distance between the current distribution of subsampled statistics of interest and the target distribution (obtained by bootstrapping the original data). Many different proposal methods are possible, but so long as there is some probability of reaching any state from any other the quality of the proposal should not be essential. What is more important is the choice of the optimization algorithm and that it have the ability to move out of local minima.

One advantage of the method as presented is that it can be applied to any problem where these steps can be implemented. Perhaps more excitingly, it provides a unified way to integrate new domain knowledge about the data by explicitly incorporating a generative model for the data and a model of missingness.

5.2 When to Use These Methods

Some problems are particularly amenable to missing data approaches which, like the one presented here, model a distribution over possible datasets that have filled in the missing data. Some problems will *not* benefit from this method; in particular, many of the scenarios that Section 1.4.2 addresses involve estimating properties of the data which are averages or variances of properties of each individual data point. That is, if our dataset consists of many x_i and the property of interest is simply an average over values $f(x_i)$ that depend only on x_i, there is likely not a need to model the behaior of the dataset under subsampling so explicitly. If we know that $f(x_i)$ is independent of the values of the other variables in the dataset ($f(x_i|x_{j\neq i}) = f(x_i)$), then the behavior of the mean of the $f(x_i)$ will behave predictably under subsampling and simpler methods will sufffice.

Conversely, for problems in which a property of the x_i (such as the distance from x_i to its nearest neighbor) depends not just on x_i but on the other $x_{j\neq i}$, the properties of interest will not behave predictably under subsampling, and many existing methods which are designed to estimate simple means and variances will not be effective. In these situations, the subsampling approach presented in this thesis is likely to be valuable.

5.3 Caveats

One key assumption of the subsampling model is that the statistic of interest can be calculated on a subset of the data. However, it is certainly possible to imagine datasets in which this is impossible. A simple example is genetic sequence alignments that are not missing whole sequences but rather individual nucleotides or sections of sequence. It is unclear how statistics like the proportion of sequences that fall into a cluster by genetic distance could be calculated without first imputing the missing nucleotides.

Another difficulty with the subsampling model is that it suffers from increased computational costs. While multiple imputation initializes a distribution of datasets, the distribution initialized by the subsampling method is roughly an order of magnitude larger. In addition, this distribution is then iteratively modified for thousands of steps; millions or more may be required for large datasets. Thus the subsampling method can take orders of magnitude longer to run than the imputation procedure. It is worth considering, in practice, whether the increased accuracy offered by the subsampling model is worth the added computational cost. When working with very large datasets, multiple imputation may be preferable.

5.4 Directions for Future Work

The effectiveness of the methods presented in this paper when applied to HIV raises the possibility that they could be useful in analyzing other datasets as well. Other infectious diseases such as Ebola have been studied with similar genetic methods in recent years [30]. The possibility that these methods could be more broadly useful in genomic surveillance is exciting; in addition, many fields other than infectious disease face challenges analyzing complex properties of incomplete datasets. In medicine, electronic medical records are

a promising but often incomplete source of data for which extracting useful information depends on the relationships between many different variables in the dataset. An even more natural application of this method is in social network analysis, where nearly all properties of interest involve interactions between nodes in the network and modeling the behavior of the network under subsampling could be incredibly valuable.

Another interesting area for future study involves the behavior of genetic distances and genetic clustering under subsampling, as seen in Section 3.4.2 and Figure 3.3. The results shown indicate that a single dataset can demonstrate substantial differences in the way its properties change from one subsample to the next; however, this variation can usually be decomposed into very distinct clusters, each exhibiting its own clear correlation between $|c_{\text{sub}} - c_{\text{obs}}|$ and error in the estimate of c. The mechanisms by which this behavior arises would be fascinating to understand and could substantially improve how methods like this one model missingness in genetic data.

Finally, while the Markov Chain Monte Carlo approach presented in this paper was ultimately impossible to implement because of an inability to calculate $|c_{\text{sub}}|$, overcoming this barrier would ultimately be quite valuable. Perhaps it is in fact possible through clever combinatorics to calculate or approximate $|c_{\text{sub}}|$. However, a more impressive result would be the ability to implement an MCMC that converged to the target distribution *without* explicitly calculating $|c_{\text{sub}}|$. In some ways, the most interesting aspect of the MCMC we attempted to implement is the more general problem it raises. Given a universe $\mathcal{U}$ of elements u_i, each of which is present at some unknown frequency (that is, perhaps there are twice as many elements u_1 as u_2), is it possible to draw elements from $\mathcal{U}$ according to some target distribution T over elements u_i? The ability to do so without exhaustive random sampling of the entire space (which may be prohibitively large) would be very valuable.

In conclusion, this thesis has presented new methods for analyzing datasets with missing data; these methods are widely applicable but are applied to and validated on genetic data from the HIV epidemic in this paper. The process of developing these approaches has illustrated the need for caution in genetic linkage analyses, demonstrated the shortcomings of several standard missing-data methods in analyzing complicated datasets, and raised more questions to address in both genetics and statistics. The results hopefully emphasize that missing data can increase the difficulties in dealing with complex datasets with many dependencies. Luckily, statistical modeling and a willingness to utilize the information that is available can help us learn a lot from even an incomplete glimpse of complicated data.

Bibliography

[1] National AIDS Coordinating Agency (Republic of Botswana). Botswana 2013 Global AIDS Response Report: Progress Report of the National Response to the 2011 Declaration of Commitments on HIV and AIDS. Technical report, 2014.

[2] Abdesslam Boutayeb. The impact of HIV/AIDS on human development in African countries. *BMC Public Health*, 9(Suppl 1):S3, 2009.

[3] R Wang, R Goyal, QH Lei, M Essex, and V De Gruttola. Sample size considerations in the design of cluster randomized trials of combination HIV prevention. *Clin Trials*, 11(309), 2014.

[4] B G Brenner, M Roger, J P Routy, D Moisi, M Ntemgwa, C Matte, J G Baril, R Thomas, D Rouleau, J Bruneau, R Leblanc, M Legault, C Tremblay, H Charest, and M A Wainberg. High rates of forward transmission events after acute/early HIV-1 infection. *J Infect Dis*, 195(7):951–959, 2007.

[5] Bluma G Brenner, Michel Roger, Daniela D Moisi, Maureen Oliveira, Isabelle Hardy, Reuven Turgel, Hugues Charest, Jean-Pierre Routy, Mark A Wainberg, et al. Transmission networks of drug resistance acquired in primary/early stage hiv infection. *AIDS (London, England)*, 22(18):2509, 2008.

[6] A J Leigh Brown, S J Lycett, L Weinert, G J Hughes, E Fearnhill, and D T Dunn. Transmission network parameters estimated from HIV sequences for a nationwide epidemic. *J Infect Dis*, 204(9):1463–1469, 2011.

[7] D Bezemer, A van Sighem, V V Lukashov, L van der Hoek, N Back, R Schuurman, C A Boucher, E C Claas, M C Boerlijst, R A Coutinho, and F de Wolf. Transmission networks of HIV-1 among men having sex with men in the Netherlands. *Aids*, 24(2):271–282, 2010.

[8] F Lewis, G J Hughes, A Rambaut, A Pozniak, and A J Leigh Brown. Episodic sexual transmission of HIV revealed by molecular phylodynamics. *PLoS Med*, 5(3):e50, 2008.

[9] E M Volz, K Koelle, and T Bedford. Viral phylodynamics. *PLoS Comput Biol*, 9(3):e1002947, 2013.

[10] T Stadler and S Bonhoeffer. Uncovering epidemiological dynamics in heterogeneous host populations using phylogenetic methods. *Philos Trans R Soc Lond B Biol Sci*, 368(1614):20120198, 2013.

[11] G E Leventhal, H F Gunthard, S Bonhoeffer, and T Stadler. Using an epidemiological model for phylogenetic inference reveals density dependence in HIV transmission. *Mol Biol Evol*, 31(1):6–17, 2014.

[12] E M Volz, S L Kosakovsky Pond, M J Ward, A J Leigh Brown, and S D Frost. Phylodynamics of infectious disease epidemics. *Genetics*, 183(4):1421–1430, 2009.

[13] G J Hughes, E Fearnhill, D Dunn, S J Lycett, A Rambaut, and A J Leigh Brown. Molecular phylodynamics of the heterosexual HIV epidemic in the United Kingdom. *PLoS Pathog*, 5(9):e1000590, 2009.

[14] E M Volz, E Ionides, E O Romero-Severson, M G Brandt, E Mokotoff, and J S Koopman. HIV-1 transmission during early infection in men who have sex with men: a phylodynamic analysis. *PLoS Med*, 10(12):e1001568; discussion e1001568, 2013.

[15] E M Volz, J S Koopman, M J Ward, A L Brown, and S D Frost. Simple epidemiological dynamics explain phylogenetic clustering of HIV from patients with recent infection. *PLoS Comput Biol*, 8(6):e1002552, 2012.

[16] R D Kouyos, V von Wyl, S Yerly, J Boni, P Taffe, C Shah, P Burgisser, T Klimkait, R Weber, B Hirschel, M Cavassini, H Furrer, M Battegay, P L Vernazza, E Bernasconi, M Rickenbach, B Ledergerber, S Bonhoeffer, and H F Gunthard. Molecular epidemiology reveals long-term changes in HIV type 1 subtype B transmission in Switzerland. *J Infect Dis*, 201(10):1488–1497, 2010.

[17] D Bezemer, N R Faria, A Hassan, R L Hamers, G Mutua, O Anzala, K Mandaliya, P Cane, J A Berkley, T F Rinke de Wit, C Wallis, S M Graham, M A Price, R A Coutinho, and E J Sanders. HIV Type 1 transmission networks among men having sex with men and heterosexuals in Kenya. *AIDS Res Hum Retroviruses*, 30(2):118–126, 2014.

[18] J O Wertheim, A J Leigh Brown, N L Hepler, S R Mehta, D D Richman, D M Smith, and S L Kosakovsky Pond. The global transmission network of HIV-1. *J Infect Dis*, 209(2):304–313, 2014.

[19] S J Little, S L Kosakovsky Pond, C M Anderson, J A Young, J O Wertheim, S R Mehta, S May, and D M Smith. Using HIV networks to inform real time prevention interventions. *PLoS One*, 9(6):e98443, 2014.

[20] JM Garcia-Calleja, E Gouws, and PD Ghys. National population based HIV prevalence surveys in sub-Saharan Africa: results and implications for HIV and AIDS estimates. *Sexually transmitted infections*, 82(suppl 3):iii64–iii70, 2006.

[21] Stef Van Buuren. *Flexible imputation of missing data*. CRC press, 2012.

[22] Donald B Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.

[23] N B Carnegie, R Wang, V Novitsky, and V De Gruttola. Linkage of viral sequences among HIV-infected village residents in Botswana: estimation of linkage rates in the presence of missing data. *PLoS Comput Biol*, 10(1):e1003430, 2014.

[24] SH Liu, G Erion, V Novitsky, and V De Gruttola. Viral Genetic Linkage Analysis in the Presence of Missing Data. *PLoS One*, (submitted for review), 2015.

[25] Joseph Felsenstein. Confidence limits on phylogenies: an approach using the bootstrap. *Evolution*, pages 783–791, 1985.

[26] DB Rubin. Multiple Imputation after 18+ years. *JASA*, 91(434):437–89, 1996.

[27] A Rambaut and NC Grassly. Seq-Gen: An application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Comput. Appl. Biosci.*, (13):235–238, 1997.

[28] WK Hastings. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, 1(57):97–109, 1969.

[29] Nikolai V Smirnov. On the estimation of the discrepancy between empirical curves of distribution for two independent samples. *Bull. Math. Univ. Moscou*, 2(2), 1939.

[30] Stephen K Gire, Augustine Goba, Kristian G Andersen, Rachel SG Sealfon, Daniel J Park, Lansana Kanneh, Simbirie Jalloh, Mambu Momoh, Mohamed Fullah, Gytis Dudas, et al. Genomic surveillance elucidates ebola virus origin and transmission during the 2014 outbreak. *Science*, 345(6202):1369–1372, 2014.