PRINCE: Accurate approximation of the copy number of tandem repeats

by

Julian Levin Booth

B.Sc., University of British Columbia, 2017

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
School of Computing Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Summer 2019

Approval

Name:	**Julian Levin Booth**
Degree:	**Master of Science (Computer Science)**
Title:	**PRINCE: Accurate approximation of the copy number of tandem repeats**
Examining Committee:	**Chair:** Keval Vora Assistant Professor
	Leonid Chindelevitch Senior Supervisor Assistant Professor
	Maxwell Libbrecht Supervisor Assistant Professor
	Kay C. Wiese Supervisor Associate Professor
	Cedric Chauve Examiner Professor Department of Mathematics
Date Defended:	**July 24, 2019**

Abstract

Variable-Number Tandem Repeats (VNTR) are genomic regions where a short sequence of DNA is repeated with no space in between repeats. While a fixed set of VNTRs is typically identified for a given species, the copy number at each VNTR varies between individuals within a species. Although VNTRs are found in both prokaryotic and eukaryotic genomes, the methodology called multi-locus VNTR analysis (MLVA) is widely used to distinguish different strains of bacteria, as well as cluster strains that might be epidemiologically related and investigate evolutionary rates. This thesis introduces PRINCE (Processing Reads to Infer the Number of Copies Efficiently), an algorithm that is able to accurately estimate the copy number of a VNTR given the sequence of a single repeat unit, two short flanking sequences and a set of short reads from a whole-genome sequence (WGS) experiment. This is a challenging problem, especially in the cases when the repeat region is longer than the expected read length. The proposed method computes a statistical approximation of the local coverage inside the repeat region. This approximation is then mapped to the copy number using a linear function whose parameters are fitted to simulated data. PRINCE was tested on the genomes of two datasets of *Mycobacterium tuberculosis* strains and was shown to be more accurate than two previous methods. An implementation of PRINCE in the Python language is freely available at `https://github.com/WGS-TB/PythonPRINCE`

Keywords: Bioinformatics; Software; Variable Number Tandem Repeats

Dedication

To my grandfather Harold Levin.

Acknowledgements

Thank you to Margaryta Vityaz and Mehrdad Mansouri for starting this project and their work on the first iteration of PRINCE. Rita also performed the tests on ExpansionHunter that are shown in this thesis.

Thank you to Mehrdad Bakhtiari for his help running adVNTR.

Thank you to the Examining Committee for lending me their time.

Thank you to Leonid Chindelevitch for supervising and teaching me over the last two years.

Thank you to the rest of the Computational Biology lab for their wisdom and kindness.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The research presented in this thesis expands on the research presented at the 18th International Workshop on Algorithms in Bioinformatics (WABI 2018) done by Mehrdad Mansouri, Julian Booth, Margaryta Vityaz, Cedric Chauve, and Leonid Chindelevitch.

1.1 Bacterial Genotyping

A major goal in epidemiology is the tracking and control of infectious disease outbreaks. Distinguishing specific strains and lineages of bacteria within a species is crucial to linking infection between patients, detecting outbreaks, and discriminating exogenous reinfection from endogenous reactivation. [67]

A bacterial strain is an intraspecific classification based on common genetic features. Therefore, it is usually sufficient to determine the bacterial strain to determine many key phenotypes. Phenotype is the manifestation of observable traits, in bacteria these may include pathogenicity, antibiotic resistance and virulence. Determining these bacterial phenotypes is key to any bacterial medical treatment or containment. Bacterial strain typing may be done using phenotyping or genotyping methods. Phenotyping methods are generally not capable of discriminating closely related strains, eg. observing colony morphology, or are too expensive and time consuming, eg. testing antibiotic susceptibility [35].

Phenotype is determined by both an organism's genome and environment. (Molecular) genotyping can predict bacterial strain and phenotype at a higher resolution, and often faster and cheaper, than phenotyping methods. Genotyping discriminates bacterial strains based on common genetic features. Many genotyping methods look at tandem repeats [25, 13, 32]. At specific genomic regions, identical or highly similar sequences of DNA are repeated in tandem (i.e. with no spaces in between repeats), the number of repeats may be highly variable between strains. Some genotyping methods (like MIRU-VNTR [67]) discriminate

strains by comparing the sizes of DNA fragments in laboratory assays. Other genotyping methods use DNA sequencing to recover actual DNA sequence, allowing direct examination of key genetic features.

1.2 Mycobacterium tuberculosis

Tuberculosis (TB) is a long studied disease caused by the bacterium *Mycobacterium tuberculosis* (MTB) that still poses a severe health and economic burden globally, particularly to low income countries. In 2017, there were roughly 10 million new cases of TB globally with 1.6 million related deaths [52].

Recent efforts to control and treat TB have focused on molecular genotyping methods [35]. MTB has a genome around 4.41 million base pairs (bp) long [10]. Whole genome sequencing at the turn of the century elucidated MTB's evolutionary history and led to increased understanding of its biology [15], providing avenues for treatment. For instance, the virulence (harmfulness) of MTB is dependant on its protein secretion system. The type 7 secretion system is a set of five membrane proteins (ESX1-5) involved in the transport of proteins across the highly impermeable cell wall. Knowing this, therapeutic possibilities arise, such as ESX1 being knocked out, as it was in an attenuated strain [2].

Antibiotic resistant TB is a public health crisis. In 2017, roughly 560 000 people contracted rifampicin resistant TB. Rifampicin is the most effective first-line TB drug. 82% of those cases were resistant to multiple antimicrobial drugs [52]. Genotyping has helped uncover microbial resistance mechansisms. Rifampicin works by binding the β-subunit of MTB RNA polymerase [7]. Resistant strains usually have mutations in the part of the gene that codes for the β-subunit, resulting in conformational changes that decrease the affinity for the drug [64]. Genotyping can also be used to identify antibiotic resistance preemptively in patients so they can be treated accordingly, preventing the future spread of resistance [51].

IS6110 restriction fragment length polymorphism (RFLP) typing [71] was used as the gold standard MTB genotyping method for more than a decade. IS6110 is an insertion element found in many copies throughout the MTB genome. By probing for these elements and using restriction enzymes to make cuts at specific sequences, RFLP typing can produce unique banding patterns per strain, thus each strain has its own "fingerprint". However, IS6110 RFLP typing is labor-intensive and time consuming, requiring large amounts of pure DNA. The method also requires the interpretation of complex banding patterns, and struggles to discriminate at low (<6) copy numbers [12].

1.3 Variable Number Tandem Repeats

A variable number tandem repeat (VNTR) is any tandemly repeating genomic loci. In MTB, 24 mycobacterial interspersed repetitive units (MIRUs) are regularly used for genotyping based on their reproducible nature and highly discriminatory typing results [67]. For instance, using MIRU-VNTR scientists were able to discriminate 78 out of 90 strains with 100% reproducibility in one experiment, while they were only able to discriminate 73 out of 90 strains with 80% reproducibility using IS6110 in another [39].

Related bacterial strains will typically have similar or identical copy numbers (CNs) at these loci [45, 43]. Multi-locus VNTR analysis (MLVA) refers to genotyping methods that use these genetic elements to discriminate strain type [72].

MIRU-VNTR are minisatellites, with each repeating motif (template) having a length between 51 and 111 bp. The term VNTR is often used to refer to only minisatellites, while the term short tandem repeats (STRs) are used to refer to microsatellites (repeats with motifs of length 1-6 bp). In reality, both minisatellites and microsatellites are VNTRs [70], however for the rest of this thesis, the acronym VNTR will refer to the longer minisatellite length repeats.

1.4 MIRU-VNTR Genotyping

MTB genotyping based on MIRU-VNTRs traditionally works by amplifying the MIRU loci with PCR using primers specific to the flanking regions of each repeat locus. The size of the amplicons varies with the number of repeat copies. The size separation step can be done using capillary [40] or gel [44] electrophoresis or nondenaturing high-performance liquid chromatography [21]. MIRU-VNTR genotyping has several advantages over IS6110-RFLP typing. It is faster, higher-throughput, and the results can be expressed as an array of CNs, facilitating easier comparisons [40, 68, 67].

In recent years, MIRU-VNTR has been a prevalent MTB genotyping method. The US CDC implemented a nationwide program for MIRU-VNTR genotyping of all MTB isolates in 2009 [1]. Meanwhile, European Union member states genotype 80% of drug-resistant isolates with MIRU-VNTR [54].

Although faster than IS6110-RFLP typing, MIRU-VNTR genotyping still requires labour-intensive screens [71]. With the rise of whole genome sequencing (WGS), powerful genotyping methods that look directly at the genome and work quickly and accurately seem imminent [59]. One possible method would be to count MIRU-VNTR CNs in the sequencing data, bypassing the need for a slow experiment limited by gel electrophoresis. This

would allow direct comparison with past PCR-based MIRU-VNTR genotyping data [77]. However, the prediction of CNs from WGS offers computational challenges based on the read sizes of current WGS technologies [35].

1.5 Sequencing Technologies

We're currently living through a "genomics revolution". Ushered in by second-generation sequencing technologies, the sequencing cost per genome has decreased at a faster rate than the complexity of microchips has increased [65].

The primary sequencing method for many years, until the early 2000s, was Sanger sequencing [58]. Sanger sequencing works by mixing trace amounts of radio-labelled dideoxynucleotides (ddNTPs) into a DNA extension reaction. When a ddNTP is incorporated into the DNA chain the reaction comes to a halt as there is no 3' hydroxyl group available to extend the phosphate backbone. Four parallel reactions each containing one type of ddNTP base are performed and the results are run on four lanes of a gel. The band locations correspond to the size of the DNA fragment and thus how many bp were added before the reaction was halted. Predictably, Sanger sequencing suffered from slow, low-throughput gel electrophoresis, expensive reagents, and difficulty automating the process [31].

The second-generation of genome sequencing technologies has altered how we do genetics research. The gamut of methods are grouped based on their high parallelization and throughput, and their short read lengths [63]. For example, the Solexa (Illumina) method works by cleaving the DNA into short segments and attaching predefined DNA sequences, called adapters, to the 5' and 3' ends. These adapters bind to complementary oligonucleotides bound to a flowcell. Clusters of clonal populations develop from the original flowcell-binding DNA strands in a PCR process known as bridge amplification. Sequencing is achieved using fluorescent dNTPs, the fluorophore occupies the 3' hydroxyl position preventing sequence extension until it is cleaved away. The modified dNTPs and enzymes are washed over the flow-cell bound clusters in cycles, each cycle adding one more base pair to the sequence. The identity of the incorporating nucleotide can be determined with a sensor by exciting the fluorophores with lasers [74]. Read lengths are limited due to dephasing; some fraction of the DNA strands in a colony will, by chance, not bond a dNTP during each cycle. Massive parallelization helps keep the signal strong amongst the noise, however, eventually dephasing causes the sequence to become uncertain [48]. Solexa based methods have a read length up to 300 bp. The Solexa method is used by Illumina's HiSeq 2000, which in 2012 could sequence a genome for $41 and had a raw error rate of 0.26% [53].

Third-generation sequencing promises longer read lengths without compromising quality

and throughput. Single-molecule sequencing (SMS) removes the need for synchronization that limits read length due to dephasing [60]. For example, Oxford Nanopore's technology sequences a single DNA molecule by binding it to a modified α-hemolysin in a lipid bilayer. A voltage is applied across the bilayer and an exonuclease cleaves nucleotides off the DNA strand. Each nucleotide disrupts the ions passing through the pore differently and thus detecting the change in current can be used to sequence the molecule [9]. However, third generation methods have a ways to go when it comes to error correction, and thus have not supplanted second-generation technologies. Without parallelization, there is no real-time consensus method to correct for read errors. However, since the error profile is likely to be uniform, it may be corrected algorithmically [60].

1.6 Problem

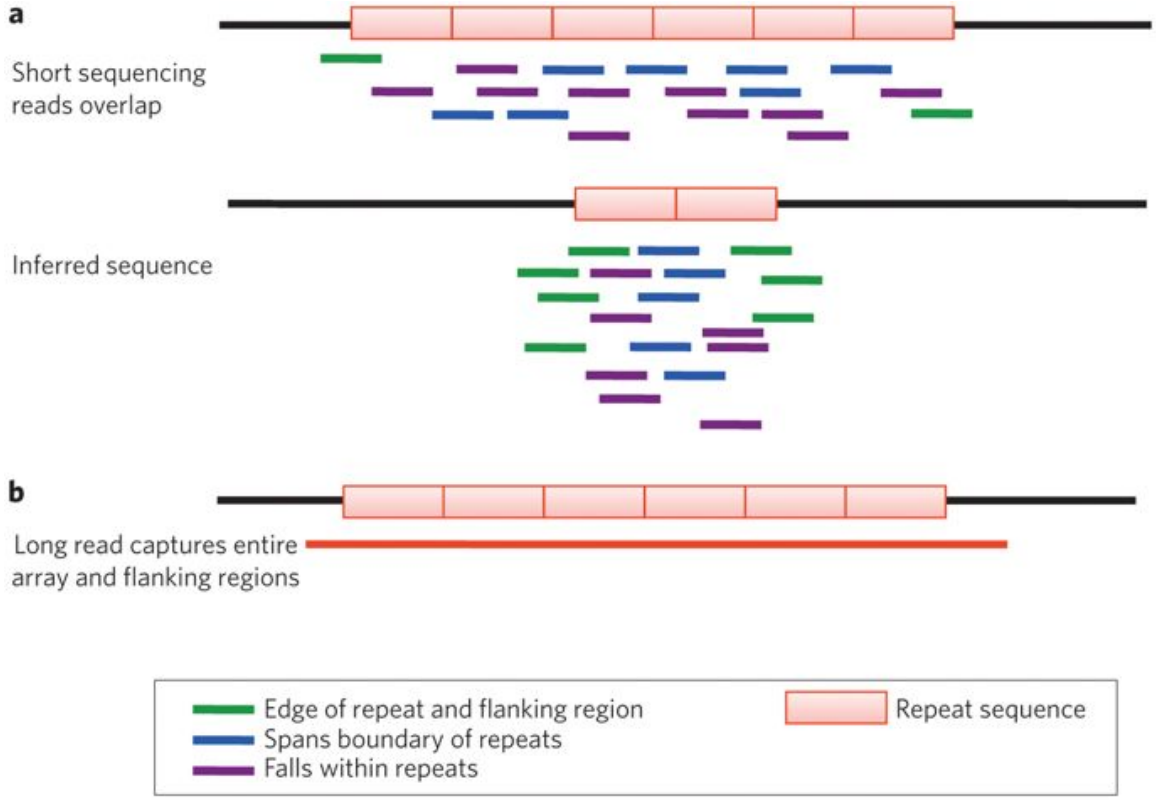
The prediction of CNs for targeted VNTR regions from whole-genome sequencing (WGS) is a well-recognized problem, that needs to be solved in order to relate bacterial strains sequenced using WGS to those genotyped in the past with PCR-based methods [35]. Unlike the previously solved problem of reconstructing spoligotypes [11], this problem presents specific challenges. The difficulty arises from the fact that in many instances, the reads produced by WGS are too short to cover the entire repeat region [27]. Instead, they only cover small sections of the repeats and cannot be assembled to reconstruct the entire, multi-copy, VNTR region [57], preventing the direct resolution of CNs (fig. 1.1). However, reads generated from the VNTR region are likely to result in an apparent higher depth of coverage of the repeated pattern compared to the average depth of coverage for the rest of the genome, a signal that can be leveraged to predict the CN.

In theory, the "coverage depth ratio" (CDR) between the repeated pattern and the rest of the genome should be roughly equal to the CN of the VNTR region. However, several factors complicate matters.

Repeat sequences can be similar to other regions within the genome, so a subsequence of a repeat sequence can appear in unrelated parts of the genome. What makes things even more challenging is that the repeated sequences may share subsequences with one another. As a result, the CDR between repeat and non-repeat regions can be highly influenced by similar, but unrelated, regions, biasing it toward higher values.

Sequencing biases, such as GC bias, can result in uneven coverage along the genome causing a skewed CDR, especially if the VNTR region is heavily biased [20, 6]. Additionally, read errors and single-nucleotide polymorphisms (SNPs) between the sample and the reference repeated sequence can cause reads to not be identified as belonging to the VNTR region,

Figure 1.1: Short read sequencing prevents assembly of repetitive arrays of DNA sequence [36].



(**a**) Genome assemblers underestimate the true number of repeat copies. (**b**) Third Generation single-molecule sequencing can span entire repeat region.

biasing the CDR toward lower values. GC bias has been controlled for in recent years with the adoption of PCR-free sequencing protocols [34, 38].

Finally, read depth fluctuates along the genome purely by chance, an effect that becomes more pronounced at lower coverage depths [42].

1.7 Previous Methods

A wide range of algorithms have also been developed for the computational genotyping of VNTRs. They generally fall into five categories: paired-end mapping, split read, read depth, de novo assembly of a genome, and combinations of the above approaches [81].

Methods that work with microsatellites typically rely on the tandem repeat to be fully contained within the read or paired-end reads, and are therefore limited by their length. Examples of such methods include lobSTR [30] and HipSTR [78]. In addition, a number of tools, such as STRViper [8] and STRait Razor [79], focus on identifying repeat templates,

rather than predicting the CN for a given template.

MIRU-profiler [54] computes alleles at the 24 standard MIRU-VNTR loci, but only from already assembled genomes. The issue is that assembling repeat regions from reads shorter than the repeat region itself is the same problem as before, but passed off to the genome assembler.

Relatively few methods have been developed to address the problem of predicting CNs of minisatellites (VNTRs) from WGS data. Currently, there are only three such methods that are broadly applicable: CNVeM [76], adVNTR [5] and ExpansionHunter [18]. The first one, CNVem, uses a probabilistic model that utilizes the inherent uncertainty of read mapping, and uses maximum likelihood to estimate locations and CNs. The second one, adVNTR [5], uses a trained Hidden Markov model (HMM) and was developed for the human genome, but a new model can be trained and run on haploid genomes. ExpansionHunter is designed to work with PCR-free WGS short-read data. It distinguishes three categories of reads: spanning, flanking and in-repeat reads in a BAM file and uses basic statistical techniques to estimate the CN from them. There was an unsuccessful attempt to apply CNVeM to the data, and thus PRINCE is only compared to adVNTR and ExpansionHunter in this paper.

Two other existing methods work in a similar context to PRINCE and are worth mentioning. The first one of these methods, TGS-TB [62], was specifically developed to work for *Mycobacterium tuberculosis*, the organism which is used as the pilot application in this work. However, it requires a minimum read length of at least 300 bp, which exceeds the most commonly used short read technologies. The final one, VNTRseek [26], is designed for detecting VNTRs, not predicting the CNs of VNTRs with known templates.

1.8 Data

PRINCE was evaluated on two different datasets: the BC dataset (PRJNA413593) and the Beijing dataset (PRJEB7281) (sample accession numbers can be found in the Supplementary Materials).

1.8.1 Beijing

The Beijing dataset [46] contains 110 genomes belonging to the Beijing lineage of *Mycobacterium tuberculosis*, with variable quality and coverage, allowing us to explore their effects on performance. The CNs for these datasets were established using standard experimental protocols for VNTR CN prediction [24] which are known to occasionally have errors. According to one study, when done with a commercial kit, they have 88% reproducibility, and have lower reproducibility when done using other methods [14]. According to another

study, some loci (templates) are more reproducible than others: reproducibility agreement rates are 98.9% and 91.3% for standard and hypervariable loci, respectively [50].

The Beijing dataset raw data (fastq files) can be found in the EMBL-EBI ENA SRA under the study accession PRJEB7281. WGS data was sequenced with an Illumina MiSeq Next Generation Sequencer.

1.8.2 BC

The BC dataset is a subset of the dataset collected in British Columbia from 2005 to 2014 [29], and contains 5 sets of high quality, high coverage WGS reads, each one containing 5 *Mycobacterium tuberculosis* samples sharing the same MIRU-VNTR pattern. This dataset allowed us to test PRINCE under optimal conditions.

The BC dataset raw data can be found in the NCBI SRA under accession number PRJNA413593. WGS data was sequenced with an Illumina HiSeq Next Generation Sequencer.

1.9 Contributions

This thesis offers a new method addressing the challenges of predicting CNs for VNTR in bacterial genomes. PRINCE (Processing Reads to Infer the Number of Copies Efficiently), accomplishes the goal of VNTR CN estimation by training a model using reads simulated from a reference genome with computationally "spiked-in" VNTRs that have known CNs. For each template, the prediction takes place in two stages, the first of which is independent of the considered reference genome and depends only on the template while the second one is reference-dependent. In the first stage, reads are recruited in a computationally efficient manner by finding exact k-mer matches to the template or flanking sequences within the reads. An approximate depth of coverage is then calculated as the ratio between the number of reads matching the template and the number matching the flanking sequences. This ratio is called the matching score. In the second stage, PRINCE fits the dependency of the known CNs on the matching score using linear regression. For an input consisting of previously unseen WGS data, the reads are recruited in the same way and the CNs are calculated from the resulting matching score using the fitted parameters.

PRINCE was tested on the BC and Beijing datasets. The CNs estimated by PRINCE are consistently closer to the true CNs than the ones predicted by ExpansionHunter and adVNTR, and its estimations remains robust for a range of coverages, read qualities and CNs. PRINCE is freely available at `https://github.com/WGS-TB/PythonPRINCE`.

Chapter 2

Methods

The objective of PRINCE is to find the CNs within VNTRs in a genome exclusively from a set of short reads, a set of templates (one per VNTR), a set of flanking sequences (two per VNTR, one at the 5' end and one at the 3' end), and a reference genome.

PRINCE was implemented in Python 2.7 and 3.6 as a command-line interface. It is available as a conda package through the Bioconda channel. PRINCE comes with the pre-trained *Mycobacterium tuberculosis* 24 MIRU-VNTR model described in this thesis.

2.1 Mathematical Problem Description

Definition 1: A *template* is the repeat unit of a VNTR, sometimes also referred to as the pattern of a VNTR. T is defined as the set of templates, i.e. $T := \{t_i\}_{i=1}^{m}$, where t_i is the i-th template and m is the number of templates.

Definition 2: The *flanking sequences* $f_i^{5'}$ and $f_i^{3'}$ are unique 50 bp sequences upstream and downstream of the i-th VNTR t_i. $f_i^{5'}$ corresponds to the sequence upstream of the i-th template, whereas $f_i^{3'}$ corresponds to its downstream sequence.

Definition 3: The *copy number* c_i is the number of times a template t_i is repeated in tandem in a genome G, possibly with errors. $\mathbf{c}$ is defined as the vector of the c_i's corresponding to each t_i in T, and write $\mathbf{c} := [c_1, \ldots, c_m]$.

Definition 4: The *match score* s_i is a proxy for the copy number of a template computed by PRINCE. $\mathbf{s}$ is defined as the vector of match scores corresponding to each t_i in T, and write $\mathbf{s} := [s_1, \ldots, s_m]$.

Definition 5: A *read* r_j is a subsequence of length L of a genome G, possibly corrupted by sequencing errors. R is defined as the set of reads, i.e. $R := \{r_j\}_{j=1}^{n}$, where n is the number of reads. A *mini-read* may also be referred to as r_j.

Given an assembled reference genome G_r for training and the template set T, PRINCE infers the parameters of a linear model for predicting copy numbers of the templates in the

set T. Using this model, given an input of a set of reads R from a genome $G \neq G_r$, PRINCE estimates the copy number vector $\mathbf{c}$, i.e. the number of times c_i that the i-th template $t_i \in T$ is repeated in tandem in the genome G, for each of the VNTRs.

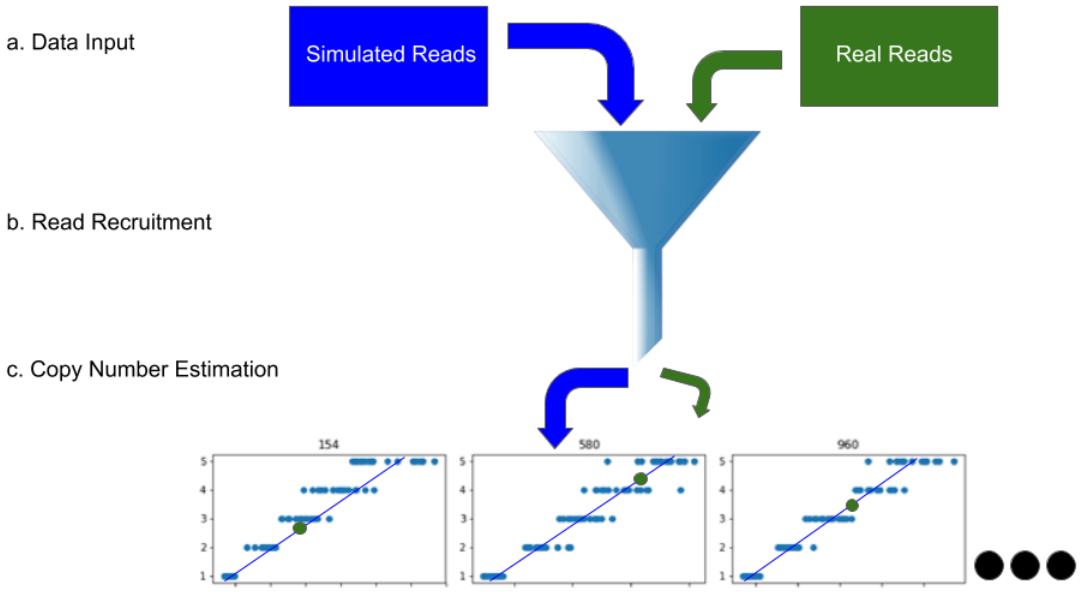
2.2 Summary of Approach

PRINCE can be thought of as the function composition $\mathbf{g}(f(R))$. Let $\mathcal{R}$ represent the domain of read sets, S represent the domain of match score vectors and C represent the domain of CN vectors. Then

$$\begin{aligned} f &: \mathcal{R} \to S \\ \mathbf{g} &: S \to C \end{aligned} \tag{2.1}$$

A different g_i is fitted to each t_i and depends on the reference genome G_r, whereas f is a recruitment and scoring algorithm that works the same way for all the templates and is independent of the reference genome G_r. The process PRINCE uses is shown in Figure 2.1.

Figure 2.1: Flowchart of PRINCE.



(**a**) Reads are divided into mini-reads and low-quality mini-reads are discarded. (**b**) Reads are compared against templates in two stages and match scores are computed. (**c**) PRINCE uses the match scores to estimate CNs via a linear regression model. 3 out of 24 *M. tuberculosis* VNTR regressions are shown here as an example.

2.3 Read Recruitment

In order to determine whether a read comes from a given template, PRINCE compares one to the other. This can be a computationally expensive task. For instance, using the Needleman-Wunsch algorithm [49] to align reads would take $O(\|LT_L\|_1)$ time per read, where T_L is the vector of template lengths. To overcome this, PRINCE divides each read into non-overlapping k-mers, which are called mini-reads. Here $k = 25$ as most commonly used read lengths are divisible by 25. If a read is not divisible by k, the remaining subsection is thrown away.

There are three reasons to split reads into mini-reads: 1. It normalizes read lengths between samples. 2. It allows for computationally efficient recruitment. 3. Low quality subsections of the read can be easily thrown out.

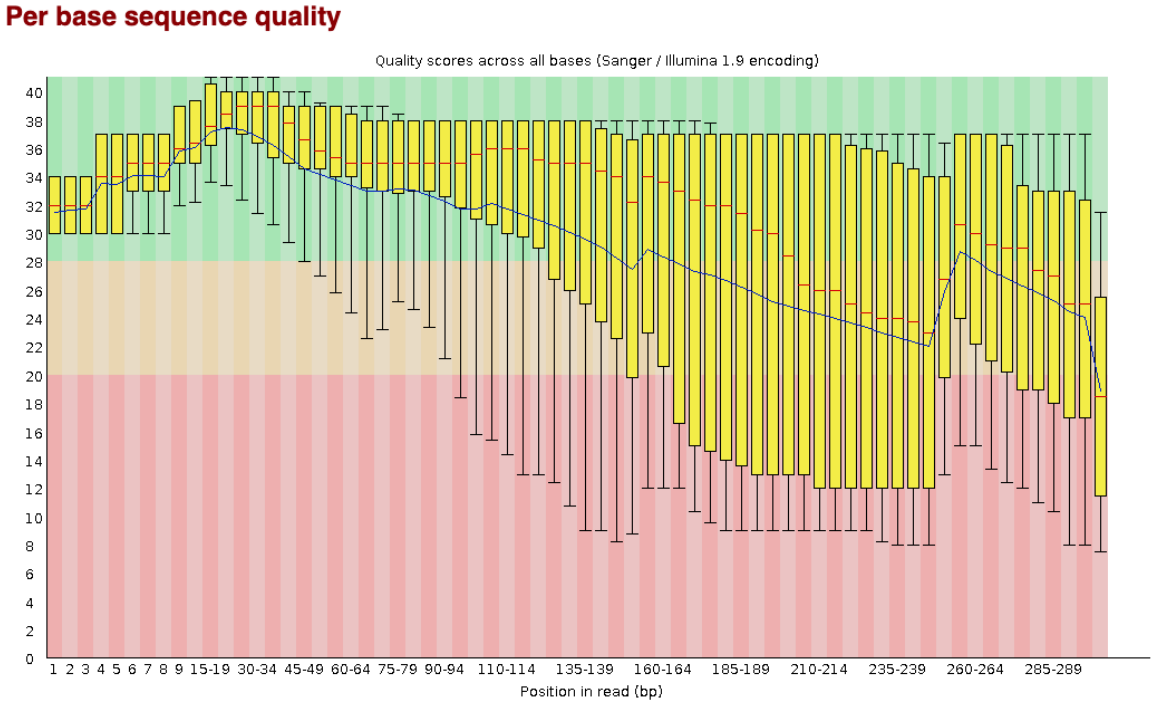
By splitting reads into mini-reads of length k, read sets that vary in read length can be processed in the same way. If R is the original read set with n reads, then the new read set $R' = \{r_j\}_{j=1}^{\lfloor \frac{nL}{k} \rfloor}$

PRINCE pre-computes the set of all unique k-mers found in each template concatenated with itself (to allow for mini-reads that fall on the boundary between two copies) and their reverse complements. For instance, if $T = \{\text{AGC, TTT}\}$, then the set of unique 2-mers of $T = \{\text{AG, GC, CA, TG, CT, TT, AA}\}$. A similar set of k-mers from the flanking sequences is also made. PRINCE then performs a k-mer matching step. In this step the mini-read is recruited as either a template read or a flanking read. If the mini-read matches one of the template k-mers, then it is recruited as a template read. If it matches a k-mer from the flanking sequences, it is recruited as a flanking read. This step is implemented using the Python set object which uses a hash table for lookups [28]. It takes time $O(1)$ to check each mini-read for membership in the sets.

To ensure that the mini-read almost certainly comes from the VNTR region, PRINCE only recruits perfect matches. No overlaps are observed between the set of template k-mers and the set of flanking k-mers in MTB. This is in line with expectations, as the flanking sequences contain the MIRU-VNTR primers, which are unique sequences for binding specificity [66]. However, if such a situation were to occur, the mini-read would be recruited as both a template read and a flanking read.

In the first step, PRINCE discards low quality mini-reads, as measured by the average Phred score [22]. A previously suggested minimum quality threshold of 20 was used [41]. Quality often degrades towards the end of the read [17]. The per-base sequence quality of

Figure 2.2: Per base sequence quality over Beijing dataset.



A box-and-whisker plot showing aggregated quality score statistics at each position along all reads in the Beijing dataset as caculated by FastQC. Y-axis is PHRED quality score. The blue line is the mean quality score at each base position. The red line within each yellow box represents the median quality score at that position. The yellow box is the inner-quartile range for 25th to 75th percentile. The upper and lower whiskers represent the 10th and 90th percentile scores.

the Beijing and BC datasets were visualized using FASTQC [4](Figures 2.2 and 2.3).

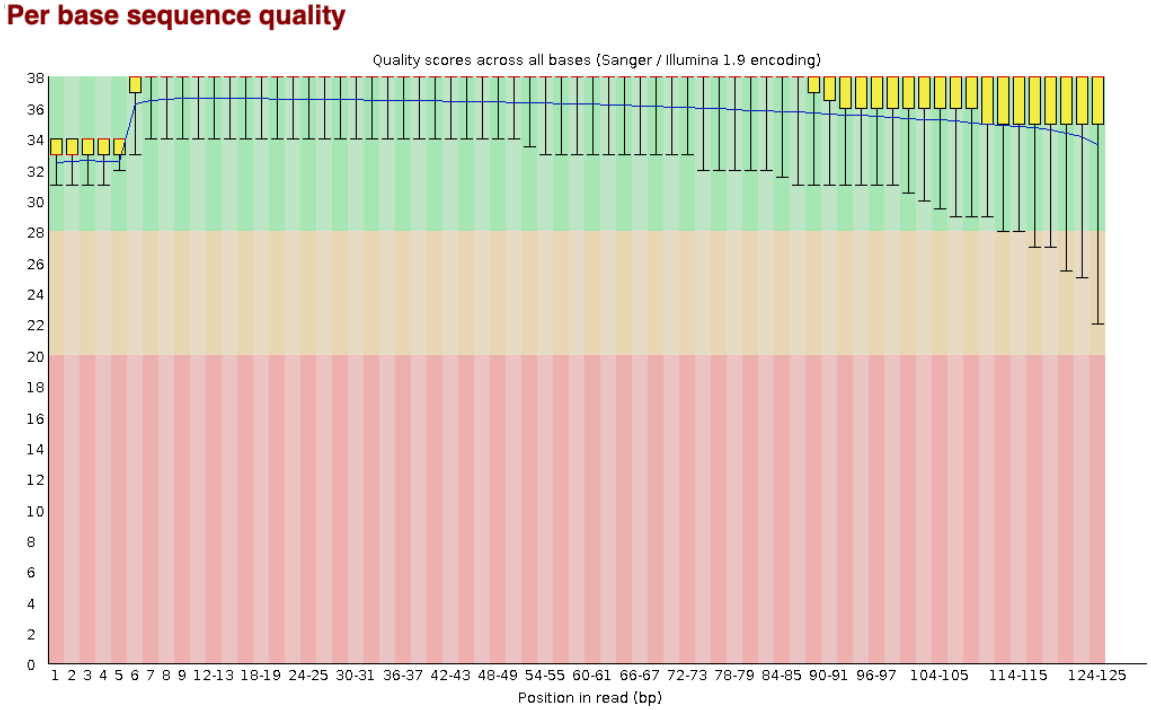
Since errors tend to cluster along the read sequence, systematic biases in quality can be efficiently dealt with by throwing away low quality mini-reads. PRINCE will keep any mini-read that meets its threshold, regardless of whether the rest of the original read was corrupted. Thus it is as if PRINCE was analyzing a higher quality, albeit lower coverage, set of reads.

2.4 Match Score Computation

The ratio of template reads to flanking reads for template t_i gives the match score s_i. Measuring flanking coverage accounts for local coverage disparities. It also ensures that all possible read sets from the same genome have the same expected match score for the same VNTR loci, no matter what the depth of coverage is.

The flanking sequences used are the primer sequences used to define MIRU-VNTRs [67]

Figure 2.3: Per base sequence quality over BC dataset.



A box-and-whisker plot showing aggregated quality score statistics at each position along all reads in the BC dataset as caculated by FastQC. Y-axis is PHRED quality score. The blue line is the mean quality score at each base position. The red line within each yellow box represents the median quality score at that position. The yellow box is the inner-quartile range for 25th to 75th percentile. The upper and lower whiskers represent the 10th and 90th percentile scores.

extended to a total of 50 bp each based on the reference genome G_r. They are extended in either direction away from the VNTR region (ie. in the 5' direction for $f_i^{5'}$). 50 bp was chosen so that enough 25-mers could be made from each sequence. 26 25-mers may be made from each of $f_i^{5'}$ and $f_i^{3'}$. If a 25 bp sequence was used, there would be too much variance in how many mini-reads actually mapped to the flanking sequence.

2.5 Parameter Fitting

PRINCE's **g** function is fitted using simulated reads, generated from the given reference genome G_r. This is done to account for all possible CNs.

Using a range of CNs at each VNTR allows PRINCE to uncover the true relationship between match score and CN. The relationship is assumed to be linear, which follows from the assumption that the coverage ratio between template and flanking reads increases linearly as more template copies are added.

PRINCE computes a vector of match scores $\mathbf{s} = f(r)$, as is described in the previous section. PRINCE fits the linear function

$$g_i(s_i) = a_i \cdot s_i + b_i = c_i \tag{2.2}$$

for each template using a linear regression [23], and uses this function to estimate CNs. By fitting the match score adjustment a_i, PRINCE becomes more robust to template-specific coverage biases such as PCR bias or substitution errors, which can be influenced by template length [20, 56]. PRINCE also accounts for the presence of homologous short sequences that may be erroneously added to s_i during read recruitment by fitting the constant b_i to be added. This assumes that the number of erroneously recruited reads recruited between samples is close to constant, an assumption that appears to hold for most of the VNTR loci given the results.

Because PRINCE only counts a mini-read towards the match score if it completely matches a template read, the first VNTR copy supplies fewer reads towards the match score than all further reads. For example, if one had constant 25x coverage along the genome and had a VNTR template length of 50 bp, then a 1 copy region would have an expected 26 mini-reads mapped to it. However, a 2 copy region would have an expected 76 mini-reads mapped to it, a 3 copy region would have 126 etc. Therefore, one may write V_i to represent the number of reads each copy after the 1st contributes to the match score and D_i to represent the default value of the first copy, B_i for the bias (i.e. erroneously recruited reads), and F_i for the number of reads recruited to $f_i^{5'}$ and $f_i^{3'}$. Then writing down the definition of the match

score as an equation and rearranging it gives

$$s_i = \frac{V_i(c_i - 1) + B_i + D_i}{F_i} \implies c_i = \frac{F_i s_i}{V_i} + 1 + \frac{-B_i - D_i}{V_i} =: a_i \cdot s_i + b_i = g_i(s_i) \quad (2.3)$$

for each template. Now it is clear that a_i actually adjusts for the number of flanking reads over the number of reads assigned to every additional VNTR copy after the first one. Meanwhile b_i is more complicated than just accounting for erroneously recruited reads.

2.6 Copy Number Prediction

Once the recruitment and scoring algorithm f returns a match score vector $\mathbf{s}$, one of two things happen, depending on whether PRINCE is being trained or queried. When training, PRINCE associates $\mathbf{s}$ with a supplied vector $\mathbf{c}$. Given a few different values for c_i at each t_i, PRINCE can fit the parameters of a linear regression, as discussed in the previous section. When querying a new set of reads with a $\mathbf{c}$ containing unknown true values, PRINCE returns a vector $\mathbf{c^*} = \mathbf{g}(\mathbf{s})$ that estimates $\mathbf{c}$. How well it estimates the true CNs will be determined in Chapter 3.

2.7 Training Pipeline

Some pre-processing must be done to train PRINCE. As implemented, training requires read sets from artificial genomes where $c_1 = c_2 = ... = c_m$ in each genome and at least two read sets R^1 and R^2 corresponding to genomes G^1 and G^2 where $c_i^1 \neq c_i^2$. This is so that there are enough data points to run a linear regression. The software to replace VNTR regions in reference genomes with fixed numbers of copies and the software necessary to artificially generate WGS files (read sets) from the artificial genomes is not currently included in the PRINCE software package.

2.7.1 Fixed Copy Insertion

Artificial genomes were made using eight different real MTB genomes listed in Appendix B [55, 47]. VNTR loci were automatically replaced using a Python script. Each region was found by searching the genome for a sequence with an edit distance of one or less from the VNTR template or its reverse complement and then removing 1000 nucleotides before and after the matching index. 1000 bp in either direction was more than enough to remove all repeat copies plus the 5' and 3' flanking sequences. Perfect matches were not found for a few VNTR/genome combinations but all VNTR regions were found using an edit distance of one or less. After the original VNTR regions were removed, the appropriate sequence was inserted. For instance at locus i, if four copies of t_i are to be inserted, then the original template region is first deleted and then $f_i^{5'} + j_i^1 + 4t_i + j_i^2 + f_i^{3'}$ is inserted, where j_i^1 and j_i^2

are randomly generated DNA sequences of length p to roughly simulate the space between the primers and the actual templates. In practice, p was set to 50 bp, but the actual distance between primers and templates varied at different loci.

As an example, if $i = 1$, t_1 = AGCTAGG, $f_1^{3'}$ = TCTAG, $f_1^{3'}$ = GCTCA, $p = 6$ and one wanted to insert three copies of t_1, then one could end up inserting into the artificial genome a sequence like **TCTAG**TAAAGA**AGCTAGGAGCTAGGAGCTAGG**TCATGC**GCTCA**, where the non-random sections of the sequence are in bold.

40 artificial genomes were made in total from 8 reference genomes. Each reference genome was made into 5 artificial genomes by removing all template regions and inserting 1 to 5 copies of all standard 24 MIRU-VNTR loci.

2.7.2 ART

Artificial WGS files were generated from the artificial genomes using the ART software [33].

ART generates realistic WGS files from assembled genomes using error model distributions extracted from empirical data.

2 artificial 125 bp paired-end read sets were generated from each artificial genome using the Illumina HiSeq 2500 profile, with 50x and 100x coverage respectively. Default settings were used, except mean fragment length was set to 200 and fragment length standard deviation was set to 10.

2.7.3 Training

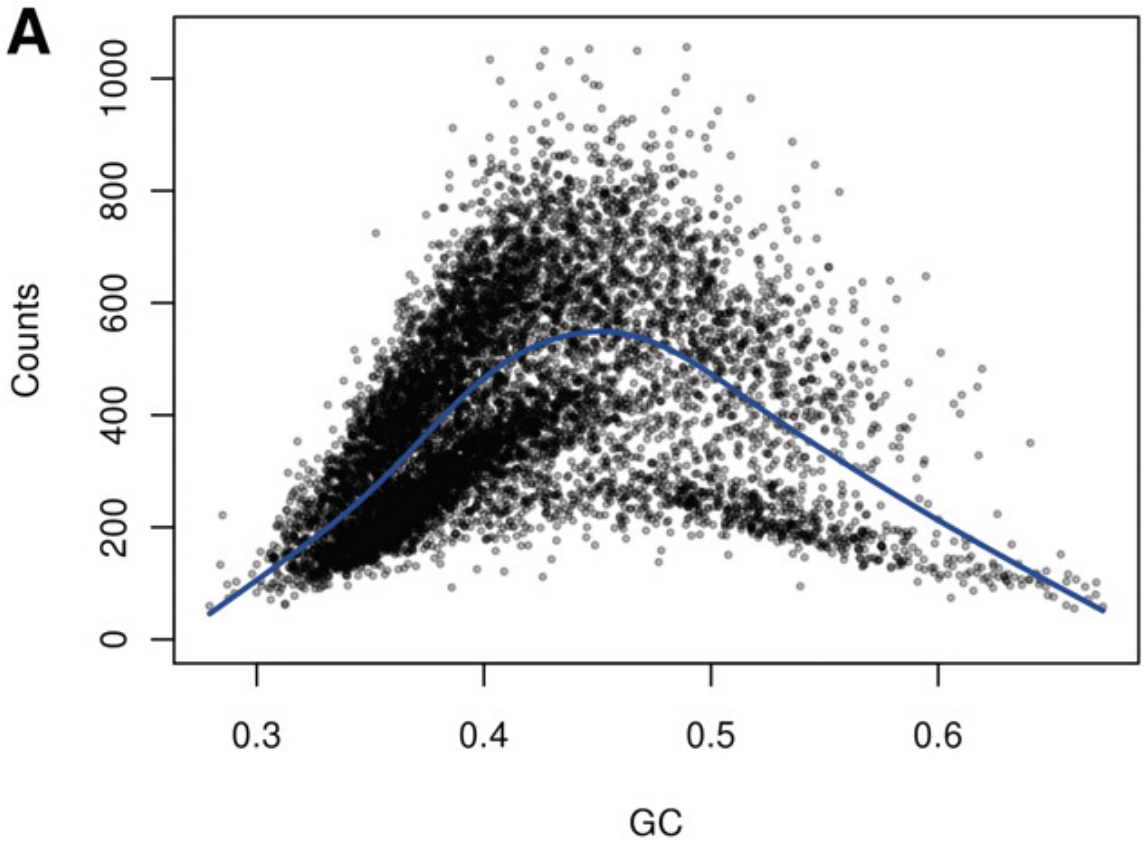
Training data for the default PRINCE MTB model was saved at https://github.com/WGS-TB/PythonPRINCE/tree/master/prince/resources/training_data.txt.

2.8 Complications

Sequencing coverage is not uniform along the genome. GC bias is introduced during the PCR amplification stage of the Solexa method used by Illumina sequencers. GC bias is a unimodal fragment level effect, fragments with low and high percentages of GC pairs have a lower depth of coverage on average [6] (Fig. 2.4). ART does not account for GC bias.

PRINCE attempts to account for GC bias by measuring read depth at both the templates and the flanking regions. Illumina fragments are generally around 250 bp, which will usually encompass both one end of the VNTR region and its closest flanking region. However, for VNTR regions with high CNs, it is likely that many DNA fragments will only cover the

Figure 2.4: Read Counts vs GC content [6].

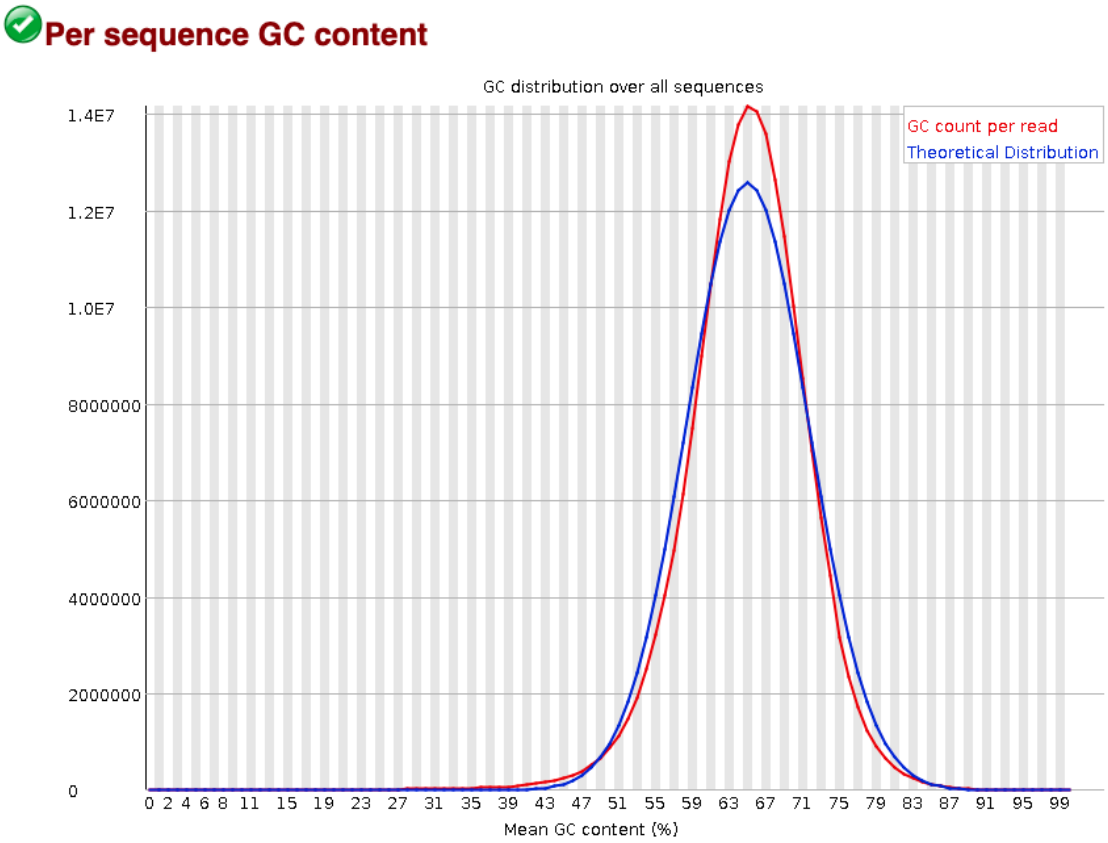


repeat region.

It is not clear whether the BC and Beijing dataset WGS files were sequenced using a PCR-free protocol. After examining the per-sequence GC content of each dataset (Figs. 2.5 and 2.6), it appears likely that GC bias has been mostly mitigated. The unimodal peaks are right near 65.6%, which is the total content in the MTB genome [10]. With a large GC bias, one would expect to see the GC content distribution shifted towards 50% as fragments with 50% GC are overrepresented in non-PCR-free Illumina sequencing (Fig. 2.4). Although that shift does appear to be there, it is extremely minor.

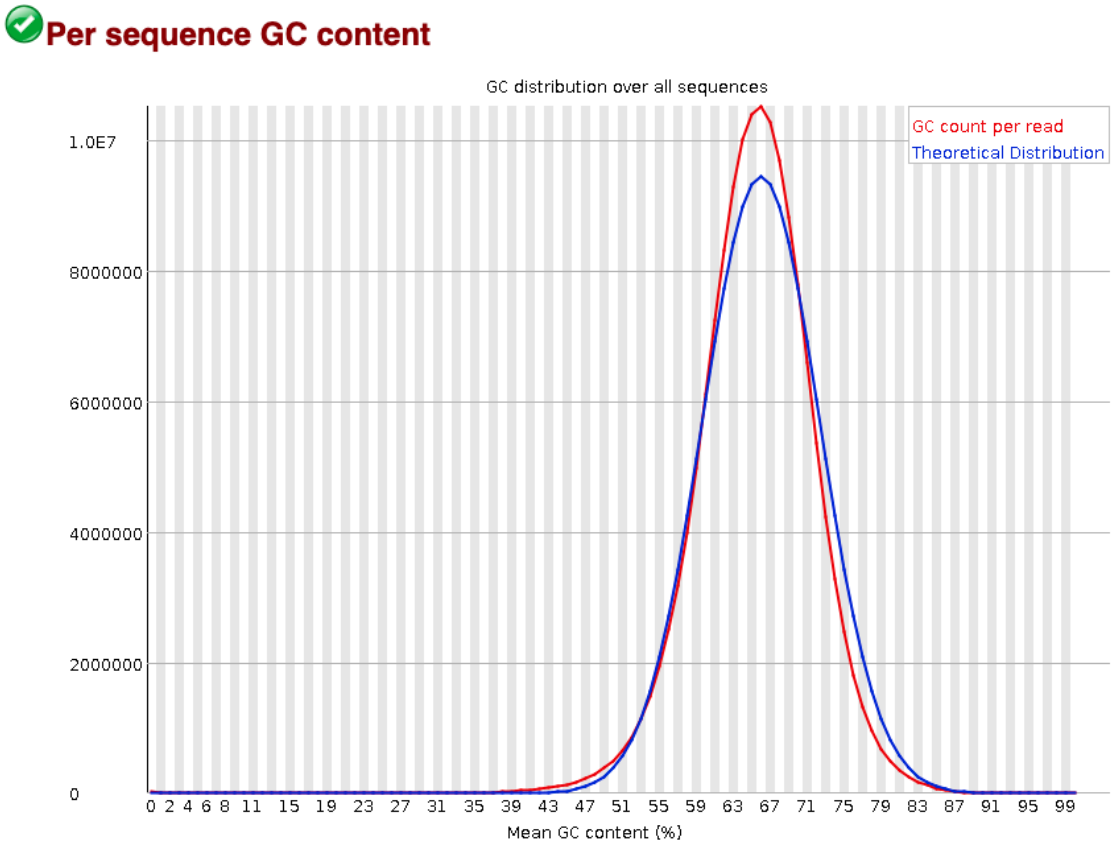
Software, such as GCcorrect [6], can be used to reduce GC bias in WGS files before using PRINCE. However, it is not included in the PRINCE pipeline, as GC bias is already controlled for in most newer WGS experiments.

Figure 2.5: Beijing Dataset GC Content.



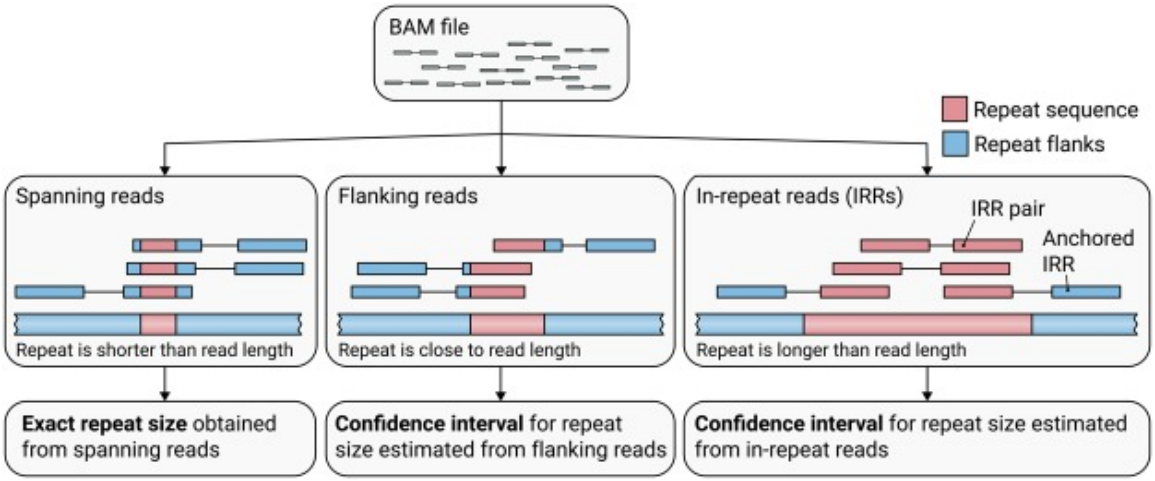
Plot of the number of reads vs. GC% per read for the Beijing Dataset as calculated by FastQC. The displayed Theoretical Distribution assumes the GC content of all reads should form a normal distribution with the peak of the curve at the mean GC content for the organism sequenced.

Figure 2.6: BC Dataset GC Content.



Plot of the number of reads vs. GC% per read for the BC Dataset as calculated by FastQC. The displayed Theoretical Distribution assumes the GC content of all reads should form a normal distribution with the peak of the curve at the mean GC content for the organism sequenced.

Figure 2.7: ExpansionHunter Methodology [18]



A summary of how ExpansionHunter estimates VNTR copy numbers starting from a binary alignment/map (BAM) file. (Left) Exact lengths of short regions are identified from spanning reads that completely contain the repeat sequence. (Middle) When the repeat length and read length are close to the same size, the copy number is approximated from the flanking reads that partially overlap the repeat and one of the repeat flanks. (Right) If the repeat region is longer than the read length, its size is estimated from reads completely contained inside the region (in-repeat reads). In-repeat reads anchored by their mate to the repeat region are used to estimate the size of the region up to the fragment length.

2.9 Other Methods

Two competing CN predicting algorithms were tested as a comparison to PRINCE.

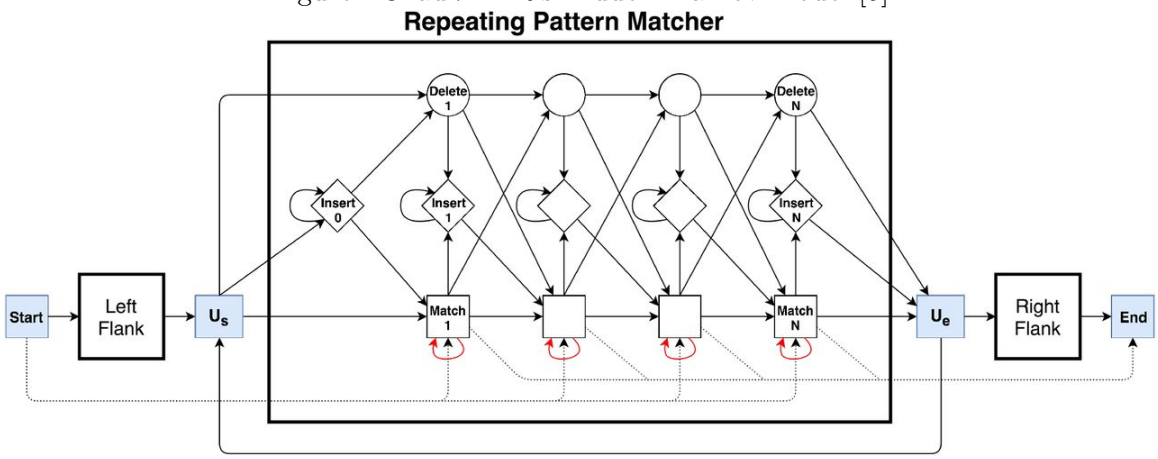
2.9.1 Expansion Hunter

ExpansionHunter will operate in one of three ways, depending on read size and repeat region size (Fig. 2.7). When the repeat region is longer than the fragment length, as is common at VNTR loci, ExpansionHunter will recruit "in-repeat" reads (IRRs) and flanking reads to estimate a lower bound on the CN. Reads are recruited if they match a template variant or its reverse complement, in a given variant catalog; mismatches are tolerated up to a certain point [18].

The unmapped reads from the Beijing and BC datasets were aligned to the Kurono genome, which was also supplied as a reference genome. For the variant catalog, one variant was used for each VNTR as found in (Fig. B)

ExpansionHunter only works for paired-end reads. Additionally, it expects PCR-free reads. Both the BC and Beijing datasets consist of paired-end reads and are unlikely to suffer from high GC bias.

Figure 2.8: adVNTR's Hidden Markov Model [5]



The adVNTR hidden Markov model (HMM) used to align reads to the flanking regions and template region. The HMM is composed of three profile HMMs, one each for the left and right flanking unique regions, and one in the middle to match multiple and partial numbers of repeat units.

2.9.2 adVNTR

adVNTR also works differently depending on read length and repeat region length. adVNTR first uses the Aho-Corasick keyword matching algorithm [3] to identify all reads that potentially match part of the VNTR or flanking regions. adVNTR then aligns the filtered reads using its trained HMM model and the Viterbi algorithm. The alignment that maximizes the output probability is used, and if the probability is higher than a specified threshold then the alignment is deemed to be valid.

Like ExpansionHunter, adVNTR then estimates a CN differently based on whether the reads can span the repeat region. When the reads are too short, adVNTR outputs a lower estimate.

adVNTR learns the transition probabilities of its HMM during training. A reference genome is supplied and a multiple sequence alignment is performed [19] based on the different repeat units found in the repeat region. The error rate of the sequencing technology is also taken into account [5] (Fig. B).

As a comparison to ExpansionHunter and PRINCE, adVNTR was trained using the Kurono genome as a reference. adVNTR was initially designed to work on diploid organisms. There is, however, a haploid setting, which was used in this analysis.

2.10 Conclusion

PRINCE, like adVNTR and ExpansionHunter, uses a two step process to estimate VNTR CNs. First the algorithm recruits relevant reads in a computationally efficient manner. Finally the algorithm uses those reads to estimate CN. In PRINCE, a read depth ratio is used, in ExpansionHunter and adVNTR, a lower estimate is provided based on how many repeat units are spanned by the reads.

While the other two methods use complex algorithms to recruit reads that may have sequencing errors, PRINCE splits reads into mini-reads and then recruits only perfectly matching mini-reads. By learning the relationship between recruited mini-reads and final CN using simulated reads, PRINCE avoids recruiting misaligned reads, but still provides consistently accurate estimations. Additionally, by using read depth to estimate CN, PRINCE can provide exact predictions for very long VNTR regions, as opposed to lower bounds.

Chapter 3

Results

The PRINCE MTB model was successfully trained on artificial read sets. PRINCE was then able to accurately estimate the VNTR values for all samples in the Beijing and BC datasets, outperforming both ExpansionHunter and adVNTR. The VNTR estimates from PRINCE were both accurate and discriminating enough to recreate known clusters in the BC dataset.

3.1 Learned Model

PRINCE was able to learn the parameters of each function in **g**. The slope and intercept of each are listed in Table 3.1. However, taking a detailed look at equation 2.3 makes sense of the results.

Slope (a_i) is equal to $\frac{F_i}{V_i}$. So $\mathbb{E}(a_i) = \mathbb{E}(F_i, \frac{1}{V_i}) = Cov(F_i, \frac{1}{V_i}) + \mathbb{E}(F_i)\mathbb{E}(\frac{1}{V_i})$. With a coverage of 25x, one should expect to see about 26 mini-reads per 50 bp flanking region, therefore $\mathbb{E}(F_i) \approx 52C$ where C is $\frac{coverage}{25}$. $\mathbb{E}(\frac{1}{V_i}) \approx \frac{1}{L_i C}$, where L_i is the length of template T_i. Therefore, $\mathbb{E}(a_i) \approx Cov(F_i, \frac{1}{V_i}) + \frac{52}{L_i}$.

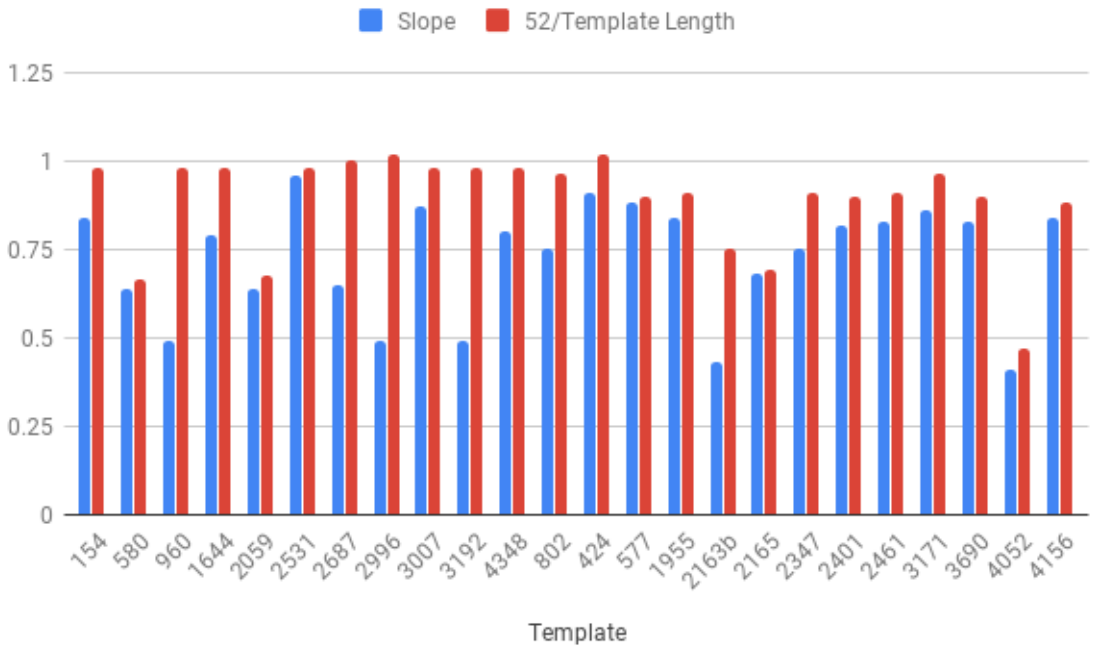
$\frac{52}{L_i}$ was larger than the actual learned slope for every loci (Figure 3.1). This can be explained by a negative covariance between F_i and $\frac{1}{V_i}$. In general the values were similar, save for a few notable exceptions. A large discrepancy does not mean that PRINCE did not learn a useful function for that VNTR, it just means that an important part was left out of equation 2.3 or an assumption was not met. PRINCE performed the worst on several of the highest discrepancy loci like 2996 and 2163b (Fig. 3.1). However, PRINCE performed relatively well on high discrepancy loci like 960 and 3192 and performed badly on high discrepancy loci like 4052. On the other hand, if the linear training model does not fit the training data well then it is not reasonable to expect that PRINCE will perform well on that locus. A linear model seemed to fit every locus well except for 2163b (Table 3.2).

Table 3.1: Learned parameters of $\mathbf{g}$

Template	Template Length (bp)	Slope a_i	Expected Slope	Intercept b_i
154	53	0.84	0.98	0.49
580	78	0.64	0.67	0.25
960	53	0.49	0.98	0.27
1644	53	0.79	0.98	0.22
2059	77	0.64	0.68	-0.26
2531	53	0.96	0.98	0.51
2687	52	0.65	1.00	0.22
2996	51	0.49	1.02	0.28
3007	53	0.87	0.98	0.69
3192	53	0.49	0.98	0.43
4348	53	0.80	0.98	0.48
802	54	0.75	0.96	0.09
424	51	0.91	1.02	0.66
577	58	0.88	0.90	0.36
1955	57	0.84	0.91	0.68
2163b	69	0.43	0.75	1.39
2165	75	0.68	0.69	0.35
2347	57	0.75	0.91	0.17
2401	58	0.82	0.90	0.57
2461	57	0.83	0.91	0.60
3171	54	0.86	0.96	-0.41
3690	58	0.83	0.90	0.68
4052	111	0.41	0.47	0.70
4156	59	0.84	0.88	0.55

For each template: it's length (bp) and the slope and intercept PRINCE learned for it, as well as its expected slope (52/length).

Figure 3.1: Slope vs $\frac{52}{L_i}$



Bar chart of the learned slope (blue) vs the expected slope (red) at each MIRU-VNTR loci.

The intercept, $b_i = 1 + \frac{-B_i - D_i}{V_i}$, which is why it is below 1, in all but one case. b_{2163b} is 1.39, likely because the number of erroneously recruited reads was not similar between training samples. As shown in Figures 3.14 and 3.15, locus 2163b was particularly difficult to estimate, both for PRINCE as well as the other methods.

3.2 Performance

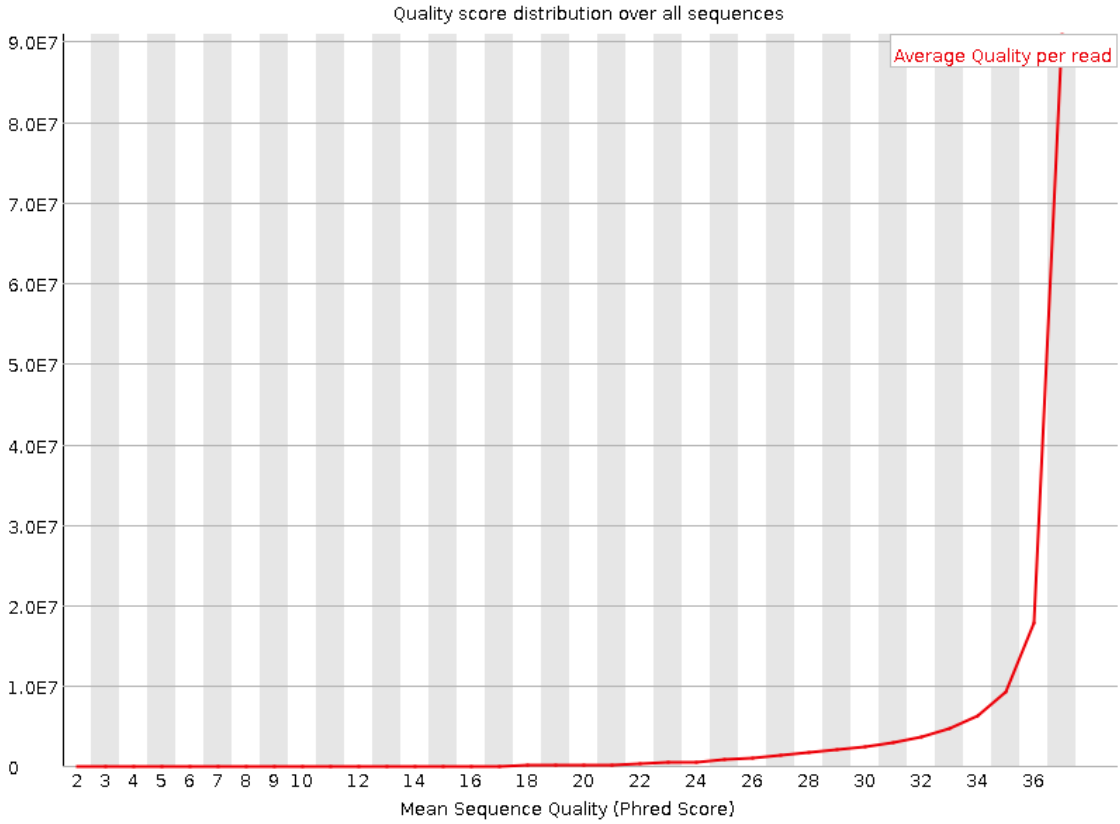
PRINCE had mean absolute errors (MAE) of 0.65 and 0.88 for the BC and Beijing datasets respectively. The stronger performance on the BC dataset could be due to the higher presence of read errors in the Beijing dataset. Dataset quality was measured by using the average Phred score per read using Fastqc [4]. The BC dataset had an average read Phred score of 35.2 (std: 3.68) (Figure 3.2), while the Beijing dataset had an average read Phred score of 31.9 (std: 6.76) (Figure 3.3). However, PRINCE performed well on lower quality datasets (Fig. 3.4). However, there was a slight increase in MAE as read quality diminished ($y = -0.0143x + 1.34$) ($R^2 = 0.124$).

Table 3.2: R-Squared Values for Learned Functions

ID	R-Squared	ID	R-Squared	ID	R-Squared
154	0.915	3007	0.922	2165	0.942
580	0.924	3192	0.817	2347	0.850
960	0.948	4348	0.947	2401	0.914
1644	0.869	802	0.885	2461	0.893
2059	0.898	424	0.931	3171	0.908
2531	0.920	577	0.930	3690	0.890
2687	0.907	1955	0.919	4052	0.820
2996	0.912	2163b	0.660	4156	0.917

R^2 measure at g_i for locus i

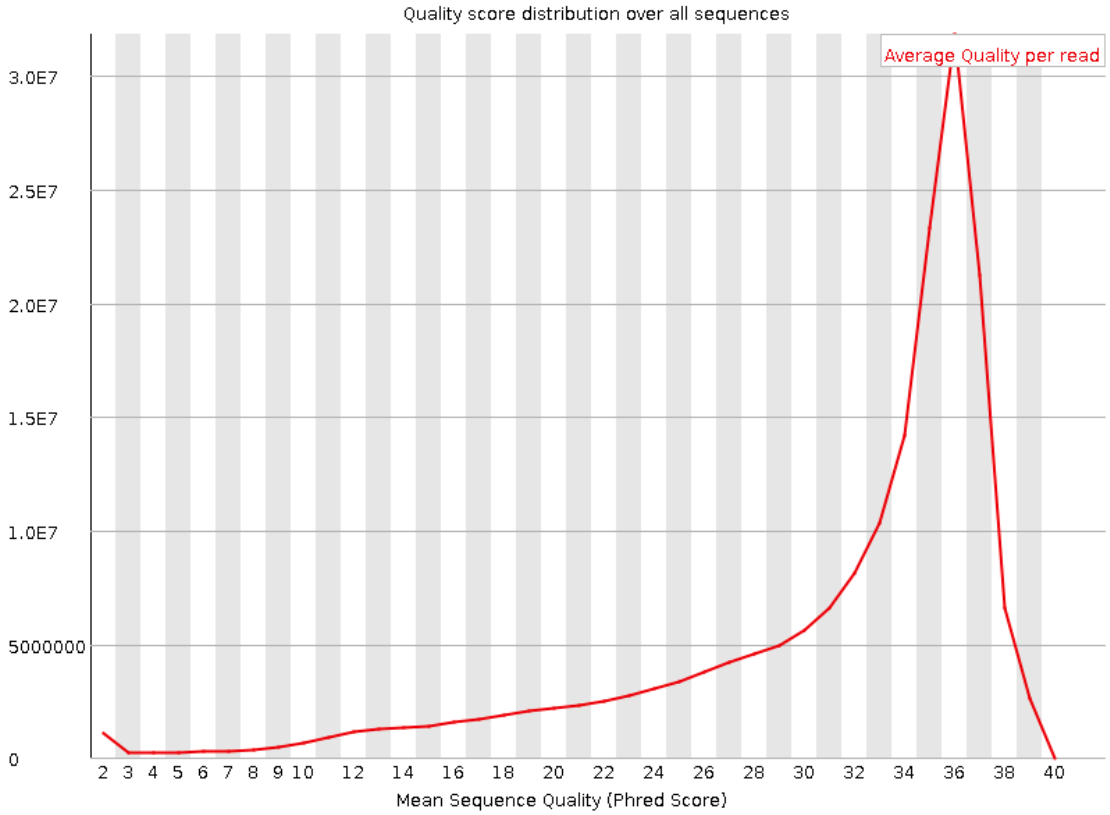
Figure 3.2: Quality Scores Distribution Over All Reads in BC Dataset



A plot of the total number of reads vs the average full length read Phred quality score for all reads in the BC dataset as calculated by FastQC.

There was no change in PRINCE's performance across different coverages ($y = -0.0002x + 0.901$) ($R^2 = 0.007$) (Fig. 3.5). Performance might be expected to sharply decline at excessively low coverages as the variance of depth along the genome becomes more significant. However, for all reasonable coverages PRINCE's performance remains consistent.

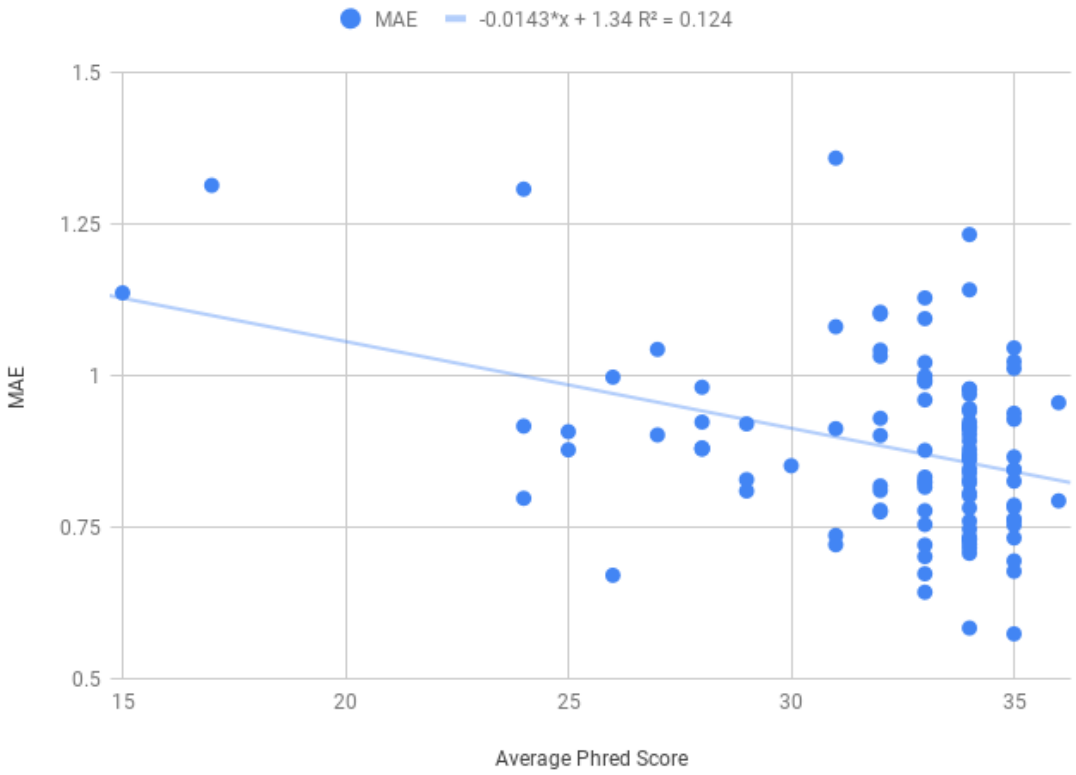
Figure 3.3: Quality Scores Distribution Over All Reads in Beijing Dataset



A plot of the total number of reads vs the average full length read Phred quality score for all reads in the Beijing dataset as calculated by FastQC.

PRINCE only recruited mini-reads that perfectly matched either the template or flanking regions. This choice was made to limit the variance in B_i between samples, at the expense of potentially throwing away mini-reads with errors or mutations that should have been recruited. PRINCE was therefore tested with a relaxed recruitment process, where mini-reads with an edit distance of ≤ 1 from a template or flanking region k-mer were recruited. For most templates, PRINCE returns better estimates when exact matching is used (Figs 3.6 and 3.7). PRINCE had an MAE of 0.77 and 0.92 for the BC and Beijing datasets respectively. Relaxing the perfect matching requirements seems to hurt performance on templates PRINCE performs well on and helps on the templates PRINCE struggles with. This is probably because perfect matching works best when model assumptions hold true and relaxed matching may aid when those assumptions are broken, specifically when VNTR repeat units don't perfectly match the templates.

Figure 3.4: MAE as a function of Phred score over the Beijing dataset.



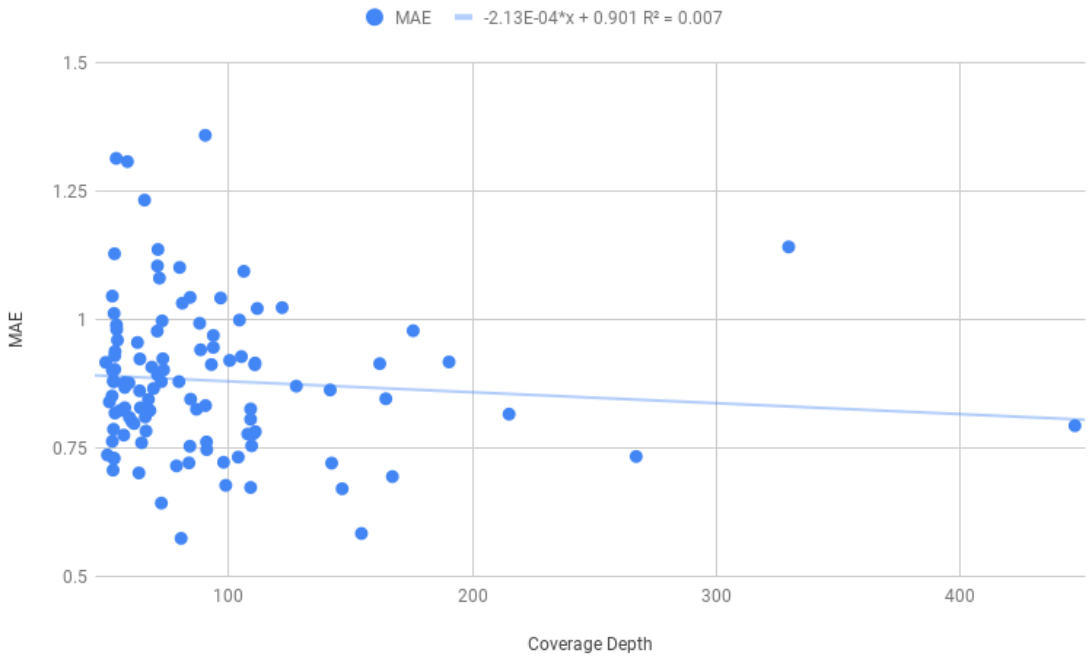
Mean average error (MAE) vs Average Phred score for each genome in the Beijing dataset.

There was a uniform discrepancy between template and flanking GC ratios (Fig. 3.8); this is because template sequences are mostly random and therefore hover around 65.6%, the average GC content of MTB [10], while flanking sequences are extended primer sequences, specifically chosen to be nearly 50% GC [16].

PRINCE partly accounts for GC bias for lower copy numbers while the sequencing fragment overlaps both the template region and a flanking region, but bias may increase as the repeat region becomes longer. There was unlikely to be much of an effect from GC bias in the datasets, and that was shown to be the case by searching for a correlation between GC content difference between flanking and template sequences versus template MAE (Fig 3.9).

PRINCE performed better on the BC dataset than on the Beijing dataset. The Beijing dataset was sequenced with an Illumina MiSeq sequencer, while the BC dataset was

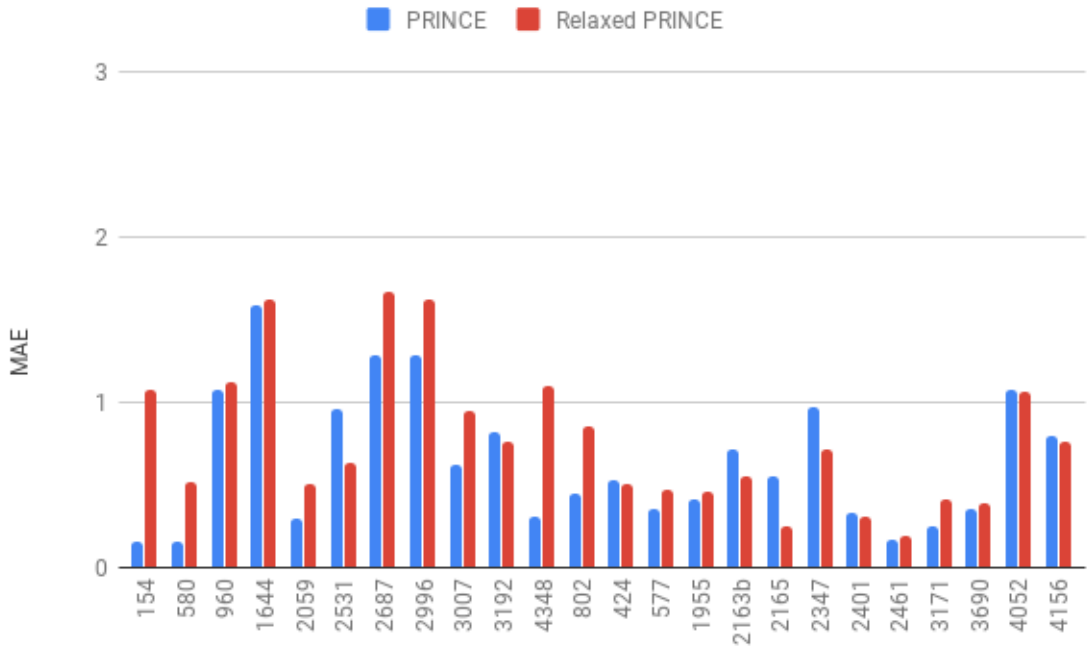
Figure 3.5: MAE as a function of coverage over the Beijing dataset.



Mean average error (MAE) vs coverage depth for each genome in the Beijing dataset.

sequenced with an Illumina HiSeq. Therefore, the Beijing dataset had longer read lengths (up to 300 bp vs 125 bp). However, the data generated by ART, used to train PRINCE, consisted of 125 bp reads generated with a HiSeq error profile. Although the read length differences shouldn't matter, due to splitting reads into mini-reads, PRINCE was still re-trained with reads reminiscent of the Beijing dataset, to see if its worse performance was due to training bias. ART was used to generate 50x and 100x WGS files from the original set of artificial genomes, but with 250 bp paired-end reads, 350 bp fragments with 10 bp standard deviation and a MiSeq V3 error profile. MiSeq-trained PRINCE had MAEs of 0.65 and 0.87 on the BC and Beijing datasets respectively compared to 0.65 and 0.88 for HiSeq-trained PRINCE. As expected, PRINCE performed better on the Beijing dataset using the artificial MiSeq data, however the difference was negligible compared to the actual difference in performance seen between the BC and Beijing datasets. Since mini-reads account for different read lengths, the main difference between the MiSeq and HiSeq versions were the error profiles. Equation 2.3 does not contain a term for read errors, flanking and template read errors should somewhat cancel each other out. The remaining differences due to read error rates should be very small as Illumina sequencing errors are rare (0.07% and 0.18% substitutions per bp for HiSeq and MiSeq technologies respectively) [61]. All subsequent results were performed using HiSeq-trained PRINCE.

Figure 3.6: MAE vs Template on BC Dataset for PRINCE with Exact Mini-read Matching and Relaxed Mini-Read Matching



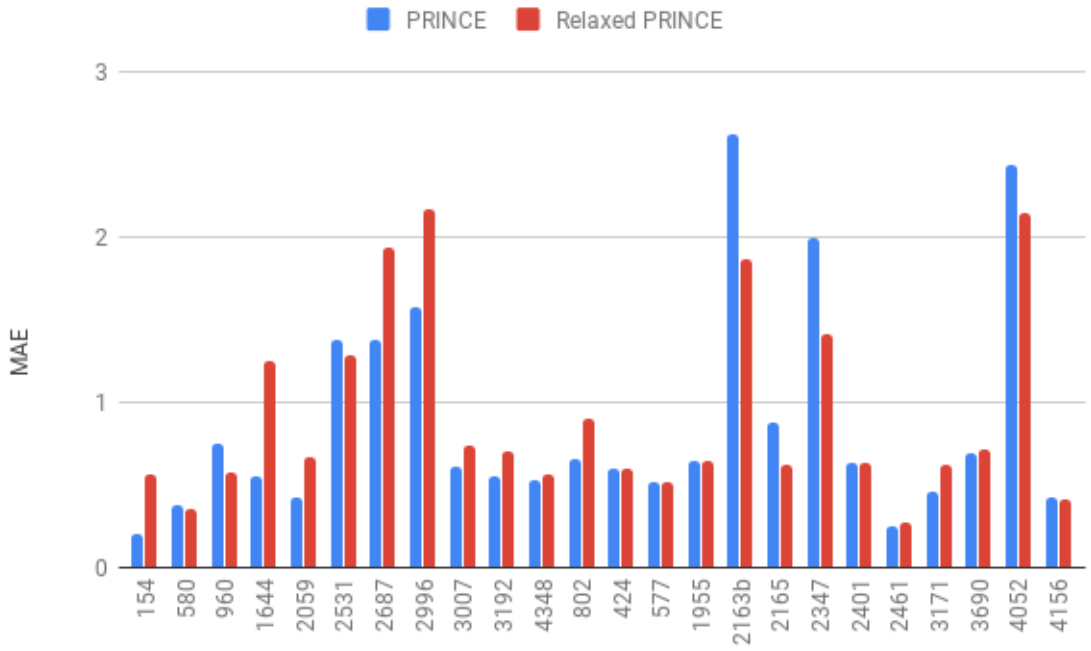
Bar chart of regular PRINCE mean average error (MAE) vs relaxed PRINCE MAE at each MIRU-VNTR loci over the BC dataset. Regular PRINCE only recruited mini-reads that exactly matched a 25-mer from the VNTR template or flanking region, while relaxed PRINCE recruited any mini-read that had an edit distance ≤ 1.

No VNTR in any genome from the two datasets used had a VNTR region with a copy number above 12. PRINCE may perform worse on higher copy numbers because g_i is trained on small copy numbers. Errors in the fitted parameter a_i are magnified as s_i increases, which is expected to happen as the number of copies increases. Although PRINCE was only trained with copy numbers from 1 to 5, PRINCE still performed well on the copy numbers seen in the testing datasets. However, PRINCE can be trained with arbitrarily high copy numbers and as few as 2 different copy numbers at each template (enough to establish a line). Training PRINCE with high copy number data may be necessary when estimating high copy number VNTR regions such as hyper-variable VNTRs [69].

3.3 Clustering

A main goal of PRINCE is to provide reliable typing results that can be used to cluster bacterial samples into related strains, information that is invaluable in epidemiology.

Figure 3.7: MAE vs Template on Beijing Dataset for PRINCE with Exact Mini-Read Matching and Relaxed Mini-read Matching

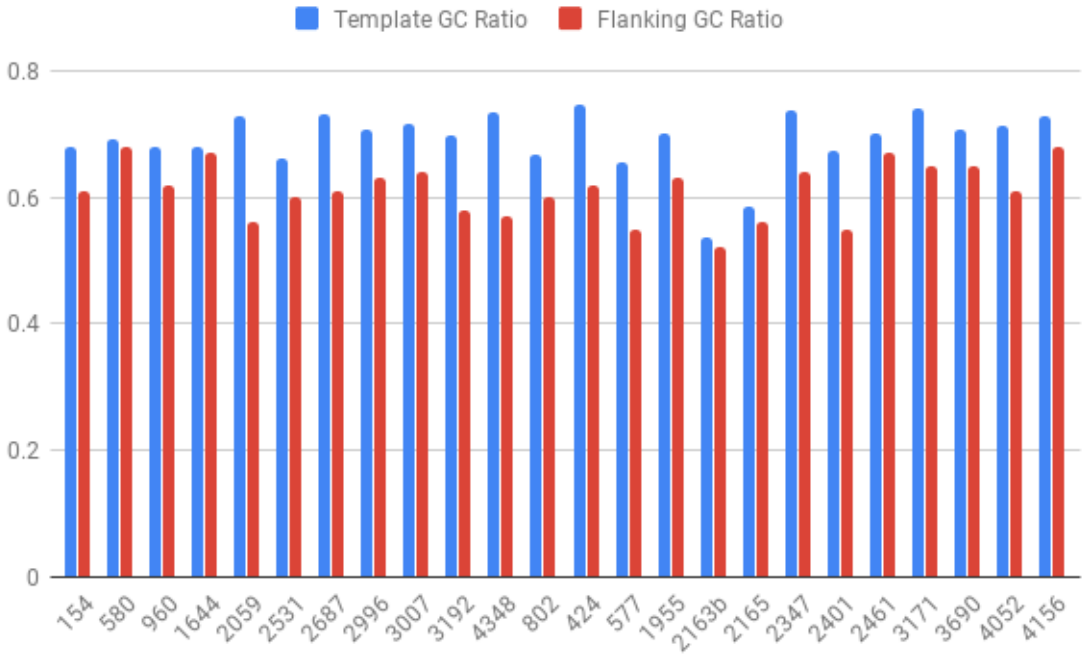


Bar chart of regular PRINCE mean average error (MAE) vs relaxed PRINCE MAE at each MIRU-VNTR loci over the Beijing dataset. Regular PRINCE only recruited mini-reads that exactly matched a 25-mer from the VNTR template or flanking region, while relaxed PRINCE recruited any mini-read that had an edit distance ≤ 1.

PRINCE was tasked with reconstructing the five sets of samples from the BC dataset. Using K-means clustering, to cluster normalized 24-dimensional copy number predictions, with $K = 5$, the original clusters were perfectly reproduced. The normalized Euclidian distances from each cluster to each centroid are listed in Table 3.3. Principal Compenent Analysis (PCA) was then used to visualize the results in two dimensions (Fig. 3.10). It is clear that PRINCE is capable of typing results discriminating enough to separate samples into their appropriate clusters, even when the data all comes from the same region.

The Beijing dataset was tested in the same way. The known clusters were themselves found in the original paper [46] using MIRU-VNTR data. The authors used a minimum spanning tree to designate 6 clonal complexes (CC1-CC6) and three distinct branches, that they collectively designated a basal sublineage (BL7) (Fig. 3.11.a).

Figure 3.8: GC Ratios of Template and Flanking Sequences



Bar chart of template GC ratio ($\frac{G+C}{A+T+G+C}$) vs flanking GC ratio for each MIRU-VNTR loci.

Again K-means clustering was used, this time with $K = 7$, on the normalized PRINCE output. While the BC dataset only has a few VNTR loci that vary between samples, and are therefore useful for genotyping, the Beijing dataset is quite variable, even between samples in the same complexes (Fig. 3.11.b).

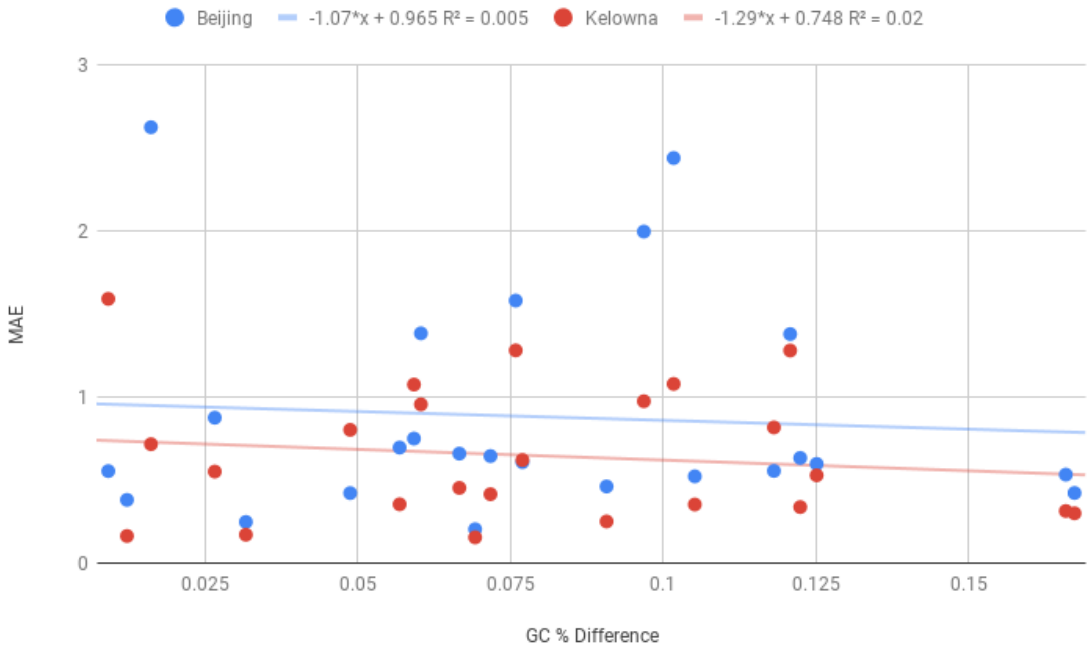
PCA was used to visualize the data, but this time three dimensions were used. In the BC dataset, the first two principal components explained 55.7% of the variance, whereas the first three components only explained 39.9% of the variance in the Beijing dataset.

Table 3.3: BC PRINCE Output K-means Clustering

Cluster/Centroid	MClust-149	MClust-035	MClust-002	MClust-007	MClust-134
MClust-149	**0.09**	0.31	0.25	0.28	0.30
MClust-035	0.30	**0.06**	0.22	0.34	0.29
MClust-002	0.26	0.24	**0.12**	0.28	0.21
MClust-007	0.28	0.35	0.27	**0.09**	0.29
MClust-134	0.31	0.31	0.21	0.29	**0.11**

Average group distance to centroid. Cell (i, j) is the average distance from group i to centroid j. Average distance of cluster to its own centroid in bold.

Figure 3.9: GC % Difference vs MAE



Scatter plot of % difference between flanking sequence GC ratio and template sequence GC ratio vs the mean average error (MAE) for each MIRU-VNTR loci averaged over the Beijing dataset (blue) and the BC dataset (red).

Even in three dimensions, the clonal complexes lie on top of each other (Fig. 3.12). The normalized Euclidian distances from each cluster to each centroid are listed in Table 3.4. However, unlike in the BC dataset, PRINCE's clusters do not perfectly correspond to the true clusters so the centroids do not correspond to anything.

The Beijing dataset consisted of a 110 sample subset of the 4987 samples used to produce the minimum spanning tree the clonal complex clusters were based on. Clustering the true CN values of the 110 sample subset with K-means also failed to separate the samples into their designated complexes (Fig. 3.13). The adjusted Rand index (ARI) for the PRINCE results clustering was 0.053 and the ARI for the true clustering was 0.12, indicating a low similarity between the k-means clusterings based on MIRU-VNTR values and the pre-determined minimum spanning tree clustered complexes.

3.4 Comparison with ExpansionHunter and adVNTR

PRINCE outperforms ExpansionHunter and adVNTR on both datasets. ExpansionHunter had an MAE of 1.62 for the BC dataset and 2.07 for the Beijing dataset, an error over twice

Table 3.4: Beijing PRINCE Output K-means Clustering

Cluster/Centroid	1	2	3	4	5	6	7
1	0.16	0.20	0.44	0.21	0.25	0.20	0.25
2	0.18	0.12	0.39	0.18	0.18	0.16	0.18
3	0.41	0.38	0.07	0.35	0.31	0.35	0.31
4	0.21	0.19	0.37	0.14	0.21	0.18	0.22
5	0.23	0.18	0.32	0.19	0.12	0.19	0.18
6	0.20	0.17	0.37	0.18	0.19	0.14	0.22
7	0.24	0.19	0.34	0.22	0.19	0.22	0.14

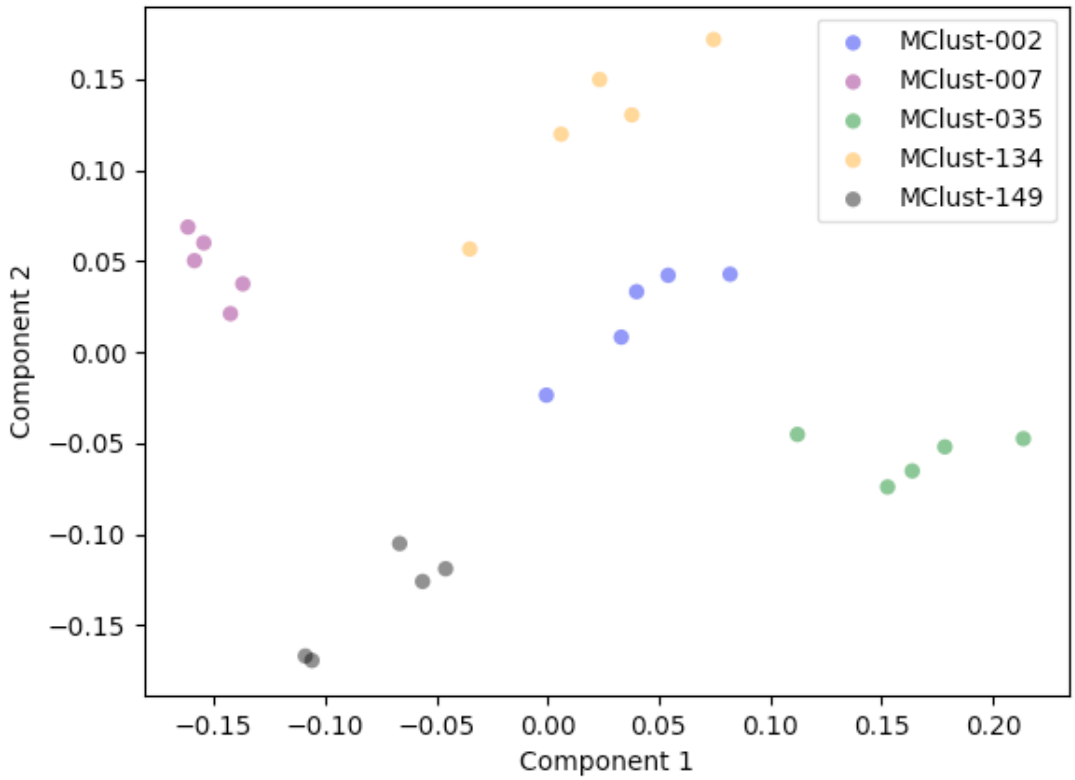
Average group distance to centroid. Cell (i, j) is the average distance from group i to centroid j.

Table 3.5: Beijing Gold Standard K-means Clustering

Cluster/Centroid	1	2	3	4	5	6	7
1	0.06	0.12	0.18	0.11	0.19	0.37	0.21
2	0.12	0.03	0.19	0.13	0.21	0.32	0.25
3	0.19	0.21	0.10	0.16	0.19	0.31	0.30
4	0.12	0.15	0.15	0.09	0.19	0.31	0.27
5	0.19	0.21	0.17	0.18	0.07	0.36	0.31
6	0.36	0.31	0.29	0.30	0.35	0.00	0.46
7	0.21	0.26	0.30	0.27	0.32	0.47	0.11

Average group distance to centroid. Cell (i, j) is the average distance from group i to centroid j.

Figure 3.10: Principal Component Analysis of BC Dataset

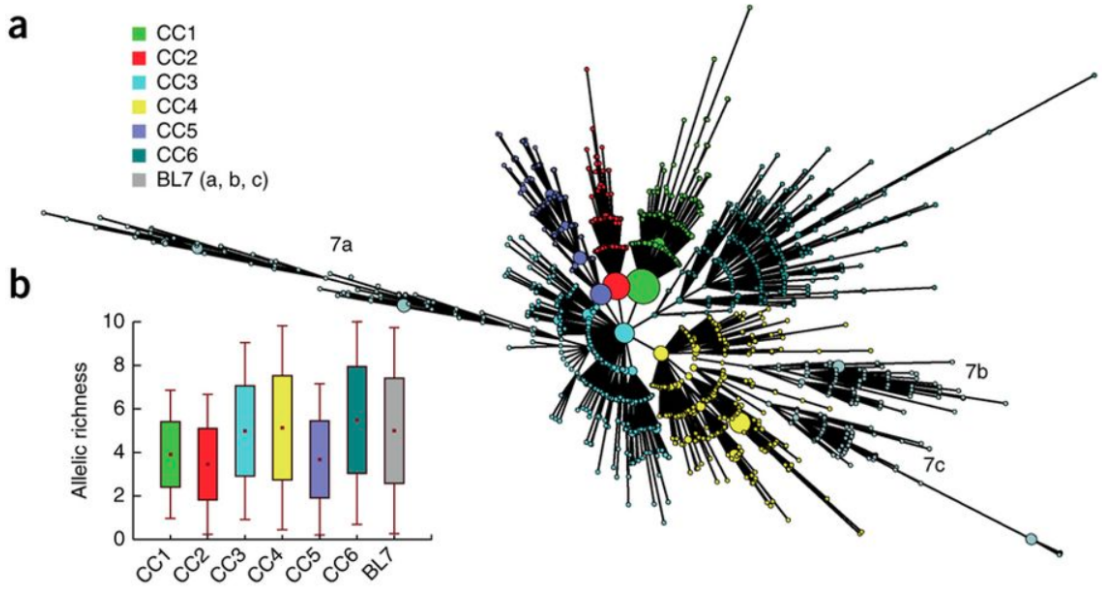


First two components of Principal Component Analysis (PCA) performed on PRINCE MIRU-VNTR estimations for the BC dataset.

that of PRINCE in each case, while adVNTR performed slightly better with MAEs of 1.10 and 1.58 over the respective datasets.

PRINCE greatly outperforms both ExpansionHunter and adVNTR on most templates in a per-template comparison on both datasets. However, on a few loci, specifically 2163b and 4052 in the Beijing dataset (MAE: 2.62 and 2.44), PRINCE gives an unusually high error (Fig. 3.14). Interestingly, all three methods perform better on 2163b in the BC dataset (PRINCE : 0.71, ExpansionHunter : 2.69, adVNTR : 1.04) (Fig. 3.15) than in the Beijing dataset (PRINCE : 2.62, ExpansionHunter : 3.70, adVNTR : 2.78). The true values for these datasets were determined using traditional typing methods. Each locus is PCR-amplified using primers that align to the flanking regions of the VNTR, the resulting PCR products are then measured by standard gel electrophoresis and a copy number is inferred based on its size [24]. The insertion of DNA between VNTR flanking regions that does not resemble the template may cause some true values to be mislabelled. The average true value for 2163b in the Beijing dataset was 5.51 compared to 3.00 in the BC dataset. The poor performance of

Figure 3.11: Beijing Dataset True Clustering [46]



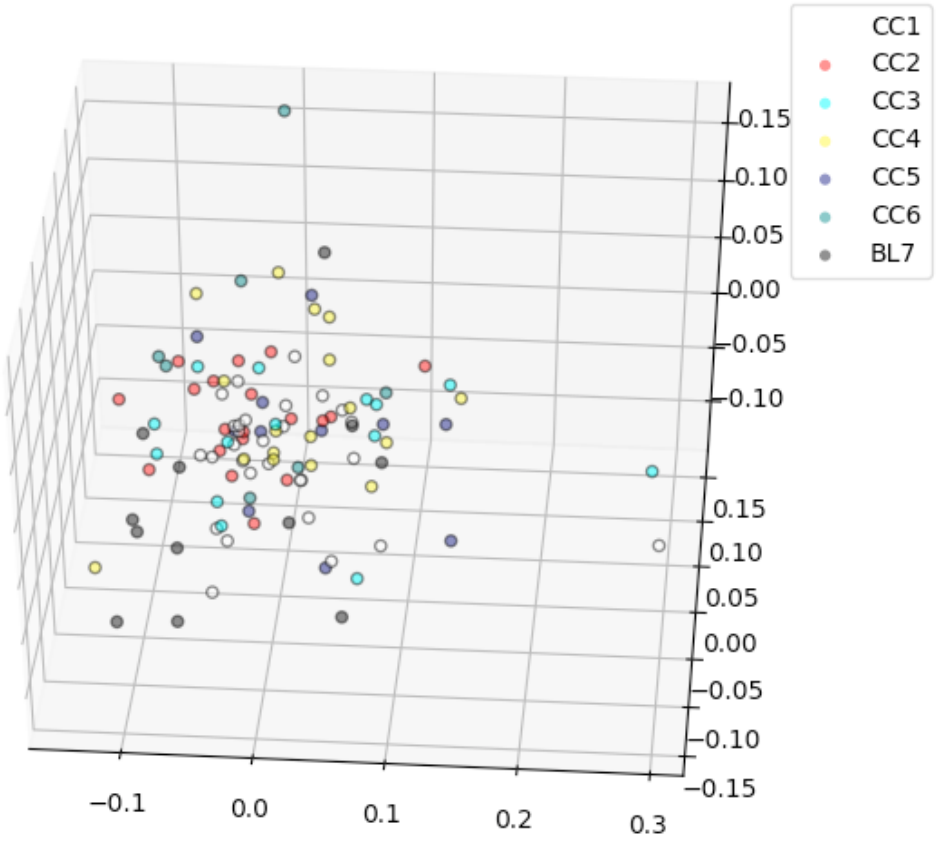
A minimum-spanning tree was used on MIRU-VNTR data to classify isolates (n=4987) into six clonal complexes and one three branch sublineage. **(a)** minimum spanning tree. **(b)** estimated allelic richness of each cluster using a rarefaction.

PRINCE, as well as ExpansionHunter and adVNTR, on 2163b might be due to incorrectly labeled true values. The same might be said for 4052, as all three methods struggled with the locus. In fact, 4052 is known to have issues with inaccurate typing [50].

PRINCE was trained with eight reference genomes, however I believe it is a fair comparison to adVNTR, which was trained on only one. adVNTR learns HMM transition probabilities from the flanking regions and repeat units of each VNTR. All of the genomes PRINCE is trained on have the exact same flanking regions and repeat units "spiked-in", training adVNTR on multiple genomes would not provide any benefit unless there were variability between repeat units and between flanking regions. However, to be sure that PRINCE isn't gaining an unfair advantage, PRINCE was trained again using only modified Kurono genomes, the important part being that PRINCE have multiple data points to perform a regression on. PRINCE had MAEs of 0.81 and 0.98 for the BC and Beijing datasets respectively, when trained just on the Kurono genome, worse than multi-genome PRINCE, but still better than the 1.10 and 1.58 produced by adVNTR.

Most of the loss in total performance comes from a drop in performance on template 3192 (Figs. 3.16 and 3.17). There appeared to be lower matching scores for the 3192 template during training with the Kurono genome. This could be explained if the number of erroneously

Figure 3.12: Principal Component Analysis of Beijing Dataset - PRINCE Output

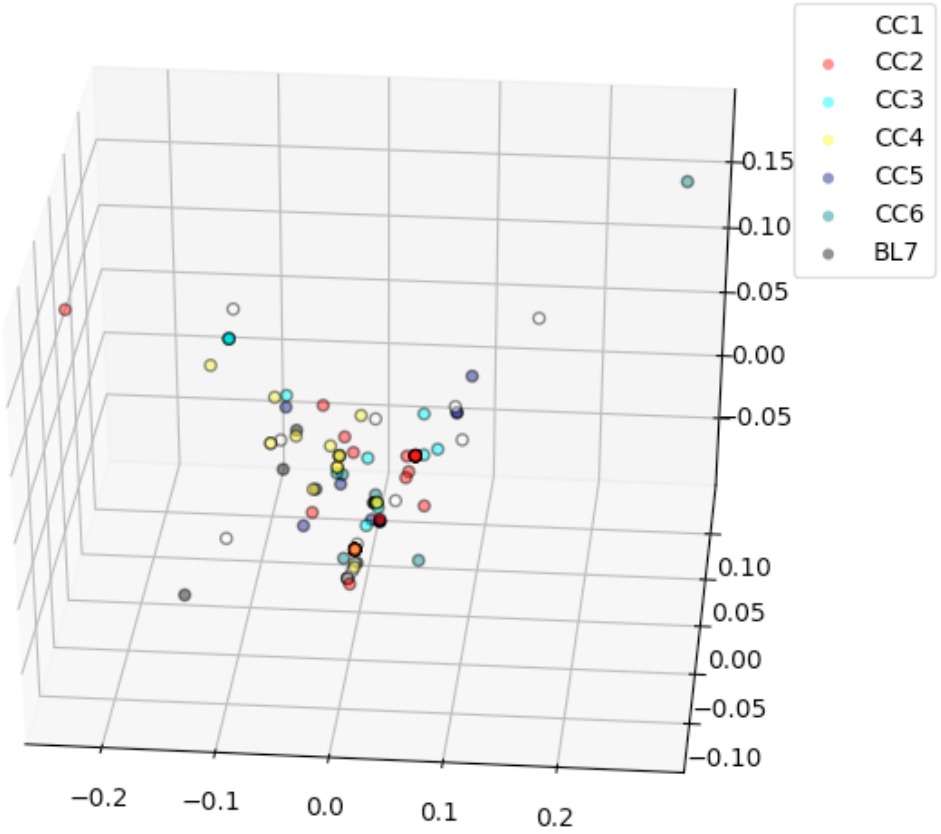


First three components of Principal Component Analysis (PCA) performed on PRINCE MIRU-VNTR estimations for the Beijing dataset.

recruited reads is higher in the other training genomes than in Kurono. NCBI's BLASTn was used to "blast" the 3192 pattern on the training genomes. All of the genomes had extra-repeat-region matches of considerable length. For instance, there is a 43 bp template substring, with 100% identity, that repeats 6 times in tandem, roughly 200 000 bp before the 3192 region. There are four copies of 3192 in both the Kurono and Beijing/391 genomes. However, 80 more 25-mers align to the Beijing/391 3192 region than the Kurono 3192 region (Fig. B). This could explain why using only the Kurono genome damages PRINCE performance for that template. If testing samples have a higher distribution of extra-repeat-region matches, then their copy numbers will be overestimated. The predictions made by Kurono-only PRINCE on 3192 were higher than the true values and the 8-genome PRINCE results for every sample. The assumption, that the number of erroneously recruited reads B_i is constant between samples, was violated, explaining the huge discrepancy between expected slope and actual slope for 3192 (Fig. 3.1)

It may be advantageous to choose a PRINCE specific set of VNTRs that are known to have

Figure 3.13: Principal Component Analysis of Beijing Dataset - MIRU-VNTR Gold Standard

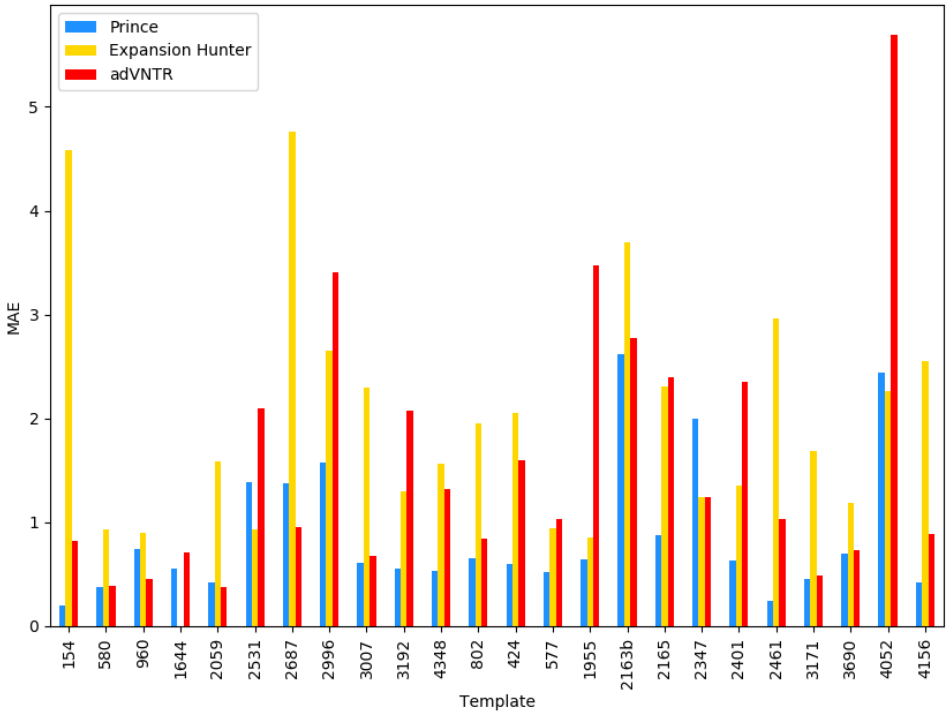


First three components of Principal Component Analysis (PCA) performed on gold standard MIRU-VNTR values for the Beijing dataset.

low extra-repeat-region match variance. This would, unfortunately, hinder the comparison between new samples and old MIRU-VNTR data. Another way to deal with extra-repeat-region matching variance could be to train PRINCE on artificial genomes made from strain specific reference genomes. For instance, if an MTB sample is known to be from the Beijing strain, then PRINCE could be trained using only Beijing strain genomes to minimize extra-repeat-region differences.

An advantage of PRINCE over the other two methods is that PRINCE outputs a fractional copy number value. ExpansionHunter and adVNTR output whole numbers, which miss out on both partial copies and the intrinsic uncertainty of inferring copy number from the variable coverage of Next Generation Sequencing data [42].

Figure 3.14: MAE vs MIRU-VNTR locus for PRINCE, ExpansionHunter and adVNTR on Beijing dataset



3.5 Running Time

With Illumina paired-end WGS data at 50x coverage, PRINCE takes an average of 75 seconds per genome to query. Training takes the same amount of time per genome; however, training may use dozens of genomes. PRINCE allows for each genome to be processed in parallel, and was trained using 80 genomes in under 5 minutes.

3.6 Conclusion

PRINCE outperformed ExpansionHunter and adVNTR on both the BC and Beijing datasets, operating over a wide range of coverage depths, read lengths and quality score distributions. PRINCE was able to provide accurate enough results on the BC dataset to perfectly cluster the samples into their original genotypes.

Figure 3.15: MAE vs MIRU-VNTR locus for PRINCE, ExpansionHunter and adVNTR on BC dataset

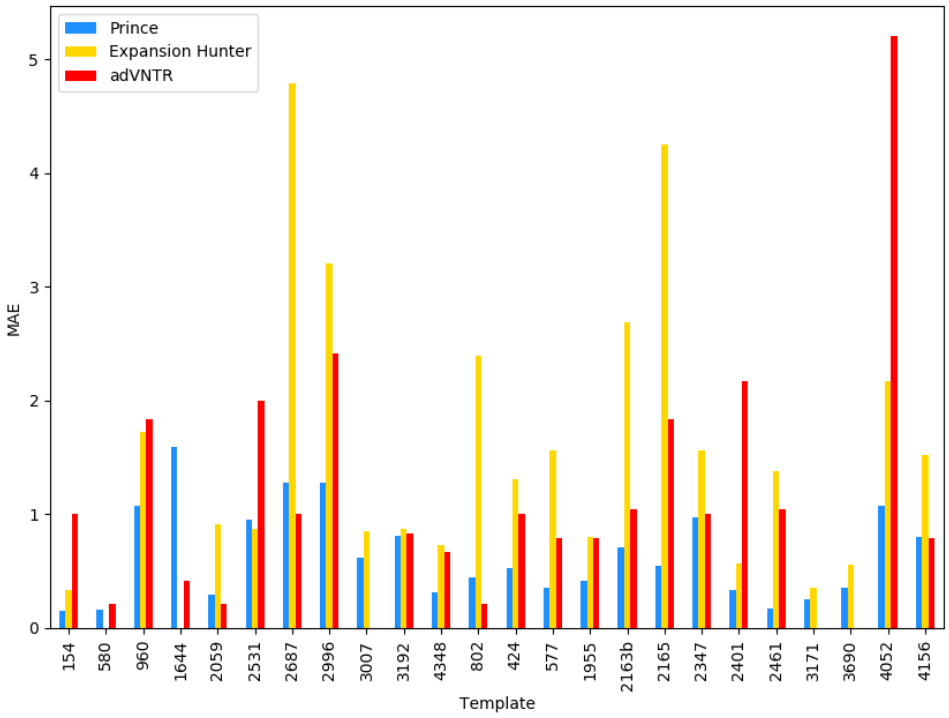
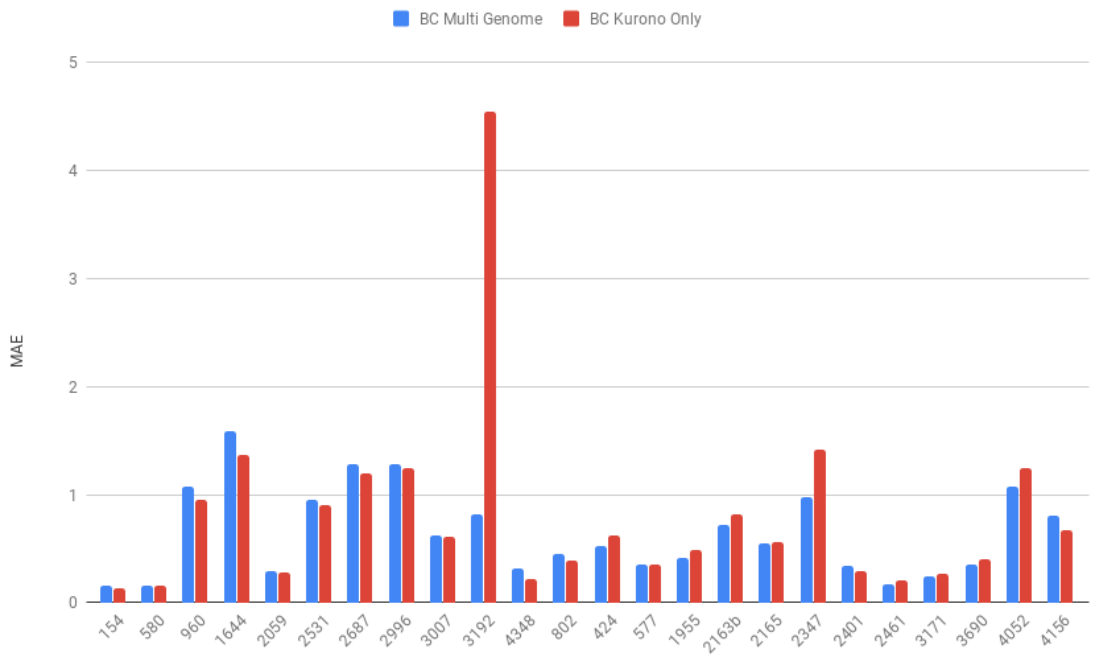
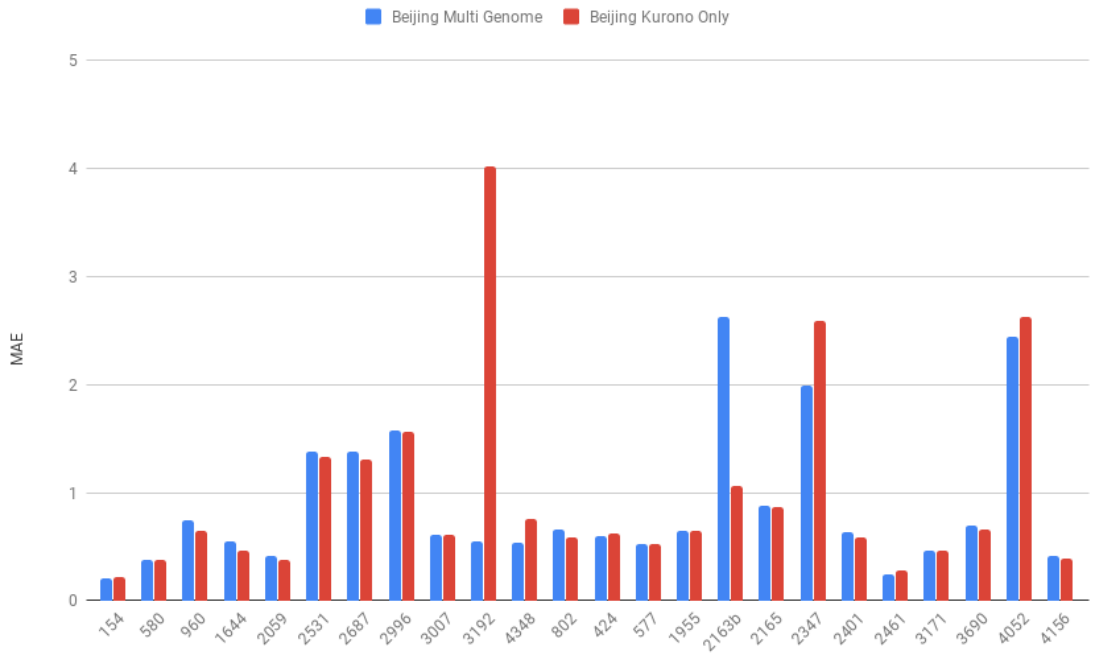


Figure 3.16: MAE vs MIRU-VNTR locus on BC Dataset for PRINCE Trained with Kurono Genome Only and All Genomes



Bar chart of mean average error (MAE) at each MIRU-VNTR locus over the BC dataset for regular PRINCE vs PRINCE trained solely using the Kurono genome as a reference genome for generating artificial reads.

Figure 3.17: MAE vs MIRU-VNTR locus on Beijing Dataset for PRINCE Trained with Kurono Genome Only and All Genomes



Bar chart of mean average error (MAE) at each MIRU-VNTR locus over the Beijing dataset for regular PRINCE vs PRINCE trained solely using the Kurono genome as a reference genome for generating artificial reads.

Chapter 4

Conclusion

4.1 Contribution

The goal of this research was to develop a fast, accurate algorithm for genotyping bacteria based on variable number tandem repeat copy numbers.

PRINCE provides an accurate estimation of the CN within a VNTR region. PRINCE fits a linear regression to the relationship between CN and the estimated depth of coverage at each locus using simulated data. *Mycobacterium tuberculosis* was used as an example of how PRINCE could be used to estimate VNTR copy numbers for any bacterial WGS data. Bacteria sequenced with WGS technology could then be compared to past samples interrogated at their VNTR loci with laboratory assays.

As input, in addition to a reference genome, PRINCE only requires a set of reads, a set of flanking sequences and a set of templates. PRINCE can hypothetically be trained on any bacterial genome, can work with reads as short as 25 base pairs, and can use any set of templates and flanking regions.

This software may open the door to a comparison between historical isolates and modern-day ones, and facilitate the extraction of novel insights on the epidemiology and transmission patterns of important bacterial pathogens such as *Mycobacterium tuberculosis*.

4.2 Future Work

PRINCE fits into a larger software package called PathOGiST. PathOGiST is an open-source algorithmic framework for clustering bacterial isolates using MLST, SNP and VNTR signals. PathOGiST clusters samples based off the individual signals before combining the clusterings using consensus clustering. PRINCE is used to retrieve the VNTR signal for MTB. Since PathOGiST is designed to operate on other pathogenic bacteria like *Escherichia*

coli [37] and *Yersinia pestis* [75], the next step would be to train PRINCE on PathOGiST compatible organisms.

The first step to training PRINCE on other organisms is identifying a set of VNTR loci. These loci must have repeat unit templates of at least 25 bp in length. In many organisms, defined VNTR sets used in MLVA include microsatellite loci. For instance, in an MLVA study on *Salmonella enterica*, 6 VNTR loci were used to type samples, 2 of the samples had template lengths of 6 and 7 bp [73], too short for PRINCE to analyze currently. In a study on *E. coli*, only 6 out of 29 putative VNTR loci had long enough template motifs for PRINCE to work [37].

A hybrid read-depth/spanning method could be a possible solution to working with microsatellites. If PRINCE called an overriding method when the read is longer than the repeat region, then PRINCE could handle most microsatellites, and perhaps improve performance on small copy number minisatellites.

While PRINCE has an advantage over ExpansionHunter, in that the former can use non-paired-end reads, leveraging paired-end information may lead to more accurate estimations, perhaps by relaxing the mini-read perfect matching requirements for reads that are paired to already recruited reads. Using paired-end information could also be used to form repeat region spanning fragments, as they are in ExpansionHunter, if a hybrid method is indeed developed.

Finally, in the future, single molecule sequencing may become commonplace. Long read technologies are much more error prone than second generation methods. The Pacific Biosciences and Oxford Nanopore Technologies platforms both have error rates of 10-20% [80]. Most of the errors are indels, unlike in Illumina sequencing where most errors are substitutions. PRINCE will need to be retrained with long read data to work with these samples. Long reads may also increase the attractiveness of a hybrid read-depth/spanning method. Longer reads would be able to span longer repeat regions, potentially as long as the longest MIRU-VNTRs. It would behoove PRINCE to keep up with changing sequencing technologies if it hopes to remain a useful tool for the foreseeable future.

Bibliography

[1] Genotyping | Data Statistics | TB | CDC.

[2] Abdallah M Abdallah, Nicolaas C Gey Van Pittius, Patricia A DiGiuseppe Champion, Jeffery Cox, Joen Luirink, Christina MJE Vandenbroucke-Grauls, Ben J Appelmelk, and Wilbert Bitter. Type VII secretion-mycobacteria show the way. *Nature Reviews Microbiology*, 5(11):883, 2007.

[3] Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

[4] S. Andrews. FASTQC. A quality control tool for high throughput sequence data, 2010.

[5] M Bakhtiari, S Shleizer-Burko, M Gymrek, V Bansal, and V Bafna. Targeted genotyping of variable number tandem repeats with adVNTR. *bioRxiv*, 2017.

[6] Yuval Benjamini and Terence P Speed. Summarizing and correcting the GC content bias in high-throughput sequencing. *Nucleic Acids Research*, 40(10):e72–e72, 2012.

[7] John S Blanchard. Molecular mechanisms of drug resistance in *Mycobacterium tuberculosis*. *Annual Review of Biochemistry*, 65(1):215–239, 1996.

[8] MD Cao, E Tasker, K Willadsen, M Imelfort, S Vishwanathan, et al. Inferring short tandem repeat variation from paired-end short reads. *Nucleic Acids Research*, 42(3):e16–e16, 2013.

[9] James Clarke, Hai-Chen Wu, Lakmal Jayasinghe, Alpesh Patel, Stuart Reid, and Hagan Bayley. Continuous base identification for single-molecule nanopore DNA sequencing. *Nature Nanotechnology*, 4(4):265, 2009.

[10] STea Cole, R Brosch, J Parkhill, T Garnier, C Churcher, D Harris, SV Gordon, K Eiglmeier, S Gas, CE Barry Iii, et al. Deciphering the biology of *Mycobacterium tuberculosis* from the complete genome sequence. *Nature*, 393(6685):537, 1998.

[11] F Coll, K Mallard, MD Preston, S Bentley, J Parkhill, et al. SpolPred: rapid and accurate prediction of *Mycobacterium tuberculosis* spoligotypes from short genomic sequences. *Bioinformatics*, 28(22):2991–2993, 2012.

[12] Lauren S Cowan, Lois Diem, Timothy Monson, Philip Wand, David Temporado, Tanya V Oemig, and Jack T Crawford. Evaluation of a two-step approach for large-scale, prospective genotyping of *Mycobacterium tuberculosis* isolates in the United States. *Journal of Clinical Microbiology*, 43(2):688–695, 2005.

[13] JW Dale, D Brittain, Angel Adrián Cataldi, D Cousins, JT Crawford, J Driscoll, H Heersma, T Lillebaek, T Quitugua, N Rastogi, et al. Spacer oligonucleotide typing of bacteria of the *Mycobacterium tuberculosis* complex: recommendations for standardised nomenclature. *The International Journal of Tuberculosis and Lung Disease*, 5(3):216–219, 2001.

[14] JL De Beer, K Kremer, C Ködmön, P Supply, D Van Soolingen, Global Network for the Molecular Surveillance of Tuberculosis 2009, et al. First worldwide proficiency study on variable-number tandem-repeat typing of *Mycobacterium tuberculosis* complex strains. *Journal of Clinical Microbiology*, 50(3):662–669, 2012.

[15] Giovanni Delogu, Michela Sali, and Giovanni Fadda. The biology of *Mycobacterium tuberculosis* infection. *Mediterranean Journal of Hematology and Infectious Diseases*, 5(1), 2013.

[16] CW Dieffenbach, TM Lowe, and GS Dveksler. General concepts for PCR primer design. *PCR Methods Appl*, 3(3):S30–S37, 1993.

[17] Juliane C Dohm, Claudio Lottaz, Tatiana Borodina, and Heinz Himmelbauer. Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. *Nucleic Acids Research*, 36(16):e105, 2008.

[18] E Dolzhenko, JJFA van Vugt, RJ Shaw, MA Bekritsky, M van Blitterswijk, et al. Detection of long repeat expansions from PCR-free whole-genome sequence data. *Genome Research*, 27(11):1895–1903, 2017.

[19] Sean R Eddy et al. Multiple alignment using hidden Markov models. In *Ismb*, volume 3, pages 114–120, 1995.

[20] M Escalona, S Rocha, and D Posada. A comparison of tools for the simulation of genomic next-generation sequencing data. *Nature Reviews Genetics*, 17(8):459, 2016.

[21] Jason T Evans, Peter M Hawkey, E Grace Smith, Kerstin A Boese, Roderic E Warren, and George Hong. Automated high-throughput mycobacterial interspersed repetitive unit typing of *Mycobacterium tuberculosis* strains by a combination of PCR and nondenaturing high-performance liquid chromatography. *Journal of Clinical Microbiology*, 42(9):4175–4180, 2004.

[22] B Ewing, L Hillier, MC Wendl, and P Green. Base-calling of automated sequencer traces using Phred. I. Accuracy assessment. *Genome Research*, 8(3):175–185, 1998.

[23] J Friedman, T Hastie, and R Tibshirani. *The Elements of Statistical Learning*, volume 1. Springer Series in Statistics New York, 2001.

[24] R Frothingham and WA Meeker-O'Connell. Genetic diversity in the *Mycobacterium tuberculosis* complex based on variable numbers of tandem DNA repeats. *Microbiology*, 144(5):1189–1196, 1998.

[25] Richard Frothingham and Winifred A Meeker-O'Connell. Genetic diversity in the *Mycobacterium tuberculosis* complex based on variable numbers of tandem DNA repeats. *Microbiology*, 144(5):1189–1196, 1998.

[26] Y Gelfand, Y Hernandez, J Loving, and G Benson. VNTRseek - a computational tool to detect tandem repeat variants in high-throughput sequencing data. *Nucleic Acids Research*, 42(14):8884–8894, 2014.

[27] S Goodwin, JD McPherson, and WR McCombie. Coming of age: ten years of next-generation sequencing technologies. *Nature Reviews Genetics*, 17(6):333–351, 2016.

[28] Micha Gorelick and Ian Ozsvald. *High Performance Python: Practical Performant Programming for Humans*. " O'Reilly Media, Inc.", 2014.

[29] JL Guthrie, C Kong, D Roth, D Jorgensen, M Rodrigues, et al. Molecular epidemiology of tuberculosis in British Columbia, Canada–a 10-year retrospective study. *Clinical Infectious Diseases*, 2017.

[30] M Gymrek, D Golan, S Rosset, and Y Erlich. lobSTR: a short tandem repeat profiler for personal genomes. *Genome Research*, 22(6):1154–1162, 2012.

[31] James M Heather and Benjamin Chain. The sequence of sequencers: The history of sequencing DNA. *Genomics*, 107(1):1–8, 2016.

[32] PW Hermans, D Van Soolingen, JW Dale, AR Schuitema, RA McAdam, D Catty, and JD Van Embden. Insertion element IS986 from *Mycobacterium tuberculosis*: a useful tool for diagnosis and epidemiology of tuberculosis. *Journal of Clinical Microbiology*, 28(9):2051–2058, 1990.

[33] W Huang, L Li, JR Myers, and GT Marth. ART: a next-generation sequencing read simulator. *Bioinformatics*, 28(4):593–594, 2012.

[34] Christopher Huptas, Siegfried Scherer, and Mareike Wenning. Optimized Illumina PCR-free library preparation for bacterial whole genome sequencing and analysis of factors influencing de novo assembly. *BMC Research Notes*, 9(1):269, 2016.

[35] T Jagielski, J van Ingen, N Rastogi, J Dziadek, PK Mazur, and J Bielecki. Current methods in the molecular typing of *Mycobacterium tuberculosis* and other mycobacteria. *BioMed Research International*, 2014(645802), 2014.

[36] Elizabeth A Kellogg. Genome sequencing: long reads for a short plant. *Nature Plants*, 1(12):15169, 2015.

[37] C Keys, S Kemper, and P Keim. Highly diverse variable number tandem repeat loci in the *E. coli* O157: H7 and O55: H7 genomes for high-resolution molecular typing. *Journal of Applied Microbiology*, 98(4):928–940, 2005.

[38] Iwanka Kozarewa, Zemin Ning, Michael A Quail, Mandy J Sanders, Matthew Berriman, and Daniel J Turner. Amplification-free Illumina sequencing-library preparation facilitates improved mapping and assembly of (G + C)-biased genomes. *Nature Methods*, 6(4):291, 2009.

[39] Kristin Kremer, Catherine Arnold, Angel Cataldi, M Cristina Gutiérrez, Walter H Haas, Stefan Panaiotov, Robin A Skuce, Philip Supply, Adri GM van der Zanden, and Dick van Soolingen. Discriminatory power and reproducibility of novel DNA typing methods for *Mycobacterium tuberculosis* complex strains. *Journal of Clinical Microbiology*, 43(11):5628–5638, 2005.

[40] Awewura Kwara, Ronald Schiro, Lauren S Cowan, Newton E Hyslop, Mark F Wiser, Stephanie Roahen Harrison, Patricia Kissinger, Lois Diem, and Jack T Crawford. Evaluation of the epidemiologic utility of secondary typing methods for differentiation of *Mycobacterium tuberculosis* isolates. *Journal of Clinical Microbiology*, 41(6):2683–2685, 2003.

[41] P Liao, GA Satten, and Y Hu. PhredEM: a Phred-score-informed genotype-calling approach for next-generation sequencing studies. *Genetic Epidemiology*, 41(5):375–387, 2017.

[42] Martin S Lindner, Maximilian Kollock, Franziska Zickmann, and Bernhard Y Renard. Analyzing genome coverage profiles with applications to quality control in metagenomics. *Bioinformatics*, 29(10):1260–1267, 2013.

[43] B Mathema, NE Kurepina, PJ Bifani, and BN Kreiswirth. Molecular Epidemiology of *Tuberculosis*: Current Insights. *Clinical Microbiology Reviews*, 19(4):658–685, 2006.

[44] Edith Mazars, Sarah Lesjean, Anne-Laure Banuls, Michele Gilbert, Véronique Vincent, Brigitte Gicquel, Michel Tibayrenc, Camille Locht, and Philip Supply. High-resolution minisatellite-based typing as a portable approach to global analysis of *Mycobacterium tuberculosis* molecular epidemiology. *Proceedings of the National Academy of Sciences*, 98(4):1901–1906, 2001.

[45] CJ Meehan, P Moris, TA Kohl, J Pečerska, S Akter, et al. The relationship between transmission time and clustering methods in *Mycobacterium tuberculosis* epidemiology. *bioRxiv*, 2018.

[46] M Merker, C Blin, S Mona, N Duforet-Frebourg, S Lecher, et al. Evolutionary history and global spread of the *Mycobacterium tuberculosis* Beijing lineage. *Nature Genetics*, 47(3):242–249, 2015.

[47] T Miyoshi-Akiyama, K Satou, M Kato, A Shiroma, K Matsumura, et al. Complete annotated genome sequence of *Mycobacterium tuberculosis* (Zopf) Lehmann and Neumann (ATCC35812)(Kurono). *Tuberculosis*, 95(1):37–39, 2015.

[48] Kensuke Nakamura, Taku Oshima, Takuya Morimoto, Shun Ikeda, Hirofumi Yoshikawa, Yuh Shiwa, Shu Ishikawa, Margaret C Linak, Aki Hirai, Hiroki Takahashi, et al. Sequence-specific error profile of Illumina sequencers. *Nucleic Acids Research*, 39(13):e90–e90, 2011.

[49] Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970.

[50] V Nikolayevskyy, A Trovato, A Broda, E Borroni, D Cirillo, and F Drobniewski. MIRU-VNTR genotyping of *Mycobacterium tuberculosis* strains using QIAxcel technology: A multicentre evaluation study. *PLoS One*, 11(3):e0149435, 2016.

[51] World Health Organization et al. The use of molecular line probe assay for the detection of resistance to isoniazid and rifampicin: policy update. 2016.

[52] World Health Organization et al. *Global tuberculosis report 2018*. World Health Organization, 2018.

[53] Michael A Quail, Miriam Smith, Paul Coupland, Thomas D Otto, Simon R Harris, Thomas R Connor, Anna Bertoni, Harold P Swerdlow, and Yong Gu. A tale of three next generation sequencing platforms: comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq sequencers. *BMC Genomics*, 13(1):341, 2012.

[54] Rahim Rajwani, Sheeba Shehzad, and Gilman Kit Hang Siu. MIRU-profiler: a rapid tool for determination of 24-loci MIRU-VNTR profiles from assembled genomes of *Mycobacterium tuberculosis*. *PeerJ*, 6:e5090, 2018.

[55] JG Rodríguez, C Pino, A Tauch, and MI Murcia. Complete genome sequence of the clinical Beijing-like strain *Mycobacterium tuberculosis* 323 using the PacBio real-time sequencing platform. *Genome Announcements*, 3(2):e00371–15, 2015.

[56] MG Ross, C Russ, M Costello, A Hollinger, NJ Lennon, et al. Characterizing and measuring bias in sequence data. *Genome Biology*, 14(5):R51, 2013.

[57] SL Salzberg and JA Yorke. Beware of mis-assembled genomes. *Bioinformatics*, 21(24):4320–4321, 2005.

[58] Frederick Sanger, Steven Nicklen, and Alan R Coulson. DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences*, 74(12):5463–5467, 1977.

[59] Giovanni Satta, M Lipman, GP Smith, C Arnold, OM Kon, and TD McHugh. *Mycobacterium tuberculosis* and whole-genome sequencing: how close are we to unleashing its full potential? *Clinical Microbiology and Infection*, 24(6):604–609, 2018.

[60] Eric E Schadt, Steve Turner, and Andrew Kasarskis. A window into third-generation sequencing. *Human Molecular Genetics*, 19(R2):R227–R240, 2010.

[61] Melanie Schirmer, Rosalinda DâĂŹAmore, Umer Z Ijaz, Neil Hall, and Christopher Quince. Illumina error profiles: resolving fine-scale variation in metagenomic sequencing data. *BMC Bioinformatics*, 17(1):125, 2016.

[62] T Sekizuka, A Yamashita, Y Murase, T Iwamoto, S Mitarai, S Kato, and M Kuroda. TGS-TB: Total genotyping solution for *Mycobacterium tuberculosis* Using Short-Read Whole-Genome Sequencing. *PLoS One*, 10(11):e0142951, 2015.

[63] Jay Shendure and Hanlee Ji. Next-generation DNA sequencing. *Nature Biotechnology*, 26(10):1135, 2008.

[64] Akos Somoskovi, Linda M Parsons, and Max Salfinger. The molecular basis of resistance to isoniazid, rifampin, and pyrazinamide in *Mycobacterium tuberculosis*. *Respiratory Research*, 2(3):164, 2001.

[65] Lincoln D Stein. The case for cloud computing in genome informatics. *Genome Biology*, 11(5):207, 2010.

[66] P Supply. Multilocus Variable Number Tandem Repeat genotyping of *Mycobacterium tuberculosis*. Technical report, Institut de Biologie/Institut Pasteur de Lille, 2005.

[67] P Supply, C Allix, S Lesjean, M Cardoso-Oelemann, S Rüsch-Gerdes, et al. Proposal for standardization of optimized mycobacterial interspersed repetitive unit-variable-number tandem repeat typing of *Mycobacterium tuberculosis*. *Journal of Clinical Microbiology*, 44(12):4498–4510, 2006.

[68] Philip Supply, Sarah Lesjean, Evgueni Savine, Kristin Kremer, Dick Van Soolingen, and Camille Locht. Automated high-throughput genotyping for study of global epidemiology of *Mycobacterium tuberculosis* based on mycobacterial interspersed repetitive units. *Journal of Clinical Microbiology*, 39(10):3563–3571, 2001.

[69] Alberto Trovato, Silva Tafaj, Simone Battaglia, Riccardo Alagna, Donika Bardhi, Perlat Kapisyzi, Silvana Bala, Migena Haldeda, Emanuele Borroni, Hasan Hafizi, et al. Implementation of a Consensus Set of Hypervariable Mycobacterial Interspersed Repetitive-Unit–Variable-Number Tandem-Repeat Loci in *Mycobacterium tuberculosis* Molecular Epidemiology. *Journal of Clinical Microbiology*, 54(2):478–482, 2016.

[70] Peter D Turnpenny and Sian Ellard. *Emery's Elements of Medical Genetics E-Book*. Elsevier Health Sciences, 2016.

[71] JD Van Embden, M Donald Cave, Jack T Crawford, JW Dale, KD Eisenach, B Gicquel, P Hermans, C Martin, R McAdam, and TM Shinnick. Strain identification of *Mycobacterium tuberculosis* by DNA fingerprinting: recommendations for a standardized methodology. *Journal of Clinical Microbiology*, 31(2):406–409, 1993.

[72] Gilles Vergnaud and Christine Pourcel. Multiple locus variable number of tandem repeats analysis. In *Molecular Epidemiology of Microorganisms*, pages 141–158. Springer, 2009.

[73] Marie-Léone Vignaud, Emeline Cherchame, Muriel Marault, Emilie Chaing, Simon Le Hello, Valerie Michel, Jourdan-Da Silva, Renaud Lailler, Anne Brisabois, Sabrina Cadel-Six, et al. MLVA for *Salmonella enterica* subsp. *enterica* serovar Dublin: development of a method suitable for inter-laboratory surveillance and application in the context of a raw milk cheese outbreak in France in 2012. *Frontiers in Microbiology*, 8:295, 2017.

[74] Karl V Voelkerding, Shale A Dames, and Jacob D Durtschi. Next-generation sequencing: from basic research to diagnostics. *Clinical Chemistry*, 55(4):641–658, 2009.

[75] Amy J Vogler, Christine E Keys, Christopher Allender, Ira Bailey, Jessica Girard, Talima Pearson, Kimothy L Smith, David M Wagner, and Paul Keim. Mutations, mutation rates, and evolution at the hypervariable VNTR loci of *Yersinia pestis*. *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis*, 616(1-2):145–158, 2007.

[76] Z Wang, F Hormozdiari, W Yang, E Halperin, and E Eskin. CNVeM: copy number variation detection using uncertainty of read mapping. *Journal of Computational Biology*, 20(3):224–236, 2013.

[77] Thomas Weniger, Justina Krawczyk, Philip Supply, Stefan Niemann, and Dag Harmsen. MIRU-VNTR plus: a web tool for polyphasic genotyping of *Mycobacterium tuberculosis* complex bacteria. *Nucleic Acids Research*, 38(suppl_2):W326–W331, 2010.

[78] T Willems, D Zielinski, J Yuan, A Gordon, M Gymrek, and Y Erlich. Genome-wide profiling of heritable and de novo STR variations. *Nature Methods*, 14(6):590, 2017.

[79] AE Woerner, JL King, and B Budowle. Fast STR allele identification with STRait Razor 3.0. *Forensic Science International: Genetics*, 30:18–23, 2017.

[80] Haowen Zhang, Chirag Jain, and Srinivas Aluru. A comprehensive evaluation of long read error correction methods. *BioRxiv*, page 519330, 2019.

[81] M Zhao, Q Wang, Q Wang, P Jia, and Z Zhao. Computational tools for copy number variation (CNV) detection using next-generation sequencing data: features and perspectives. *BMC Bioinformatics*, 14(11):S1, 2013.

Appendix A

Code

Code can be found at `https://github.com/WGS-TB/PythonPRINCE`.
Relevant files can be found in the bin/ and prince/ directories.

Appendix B

Supplemental Figures and Tables

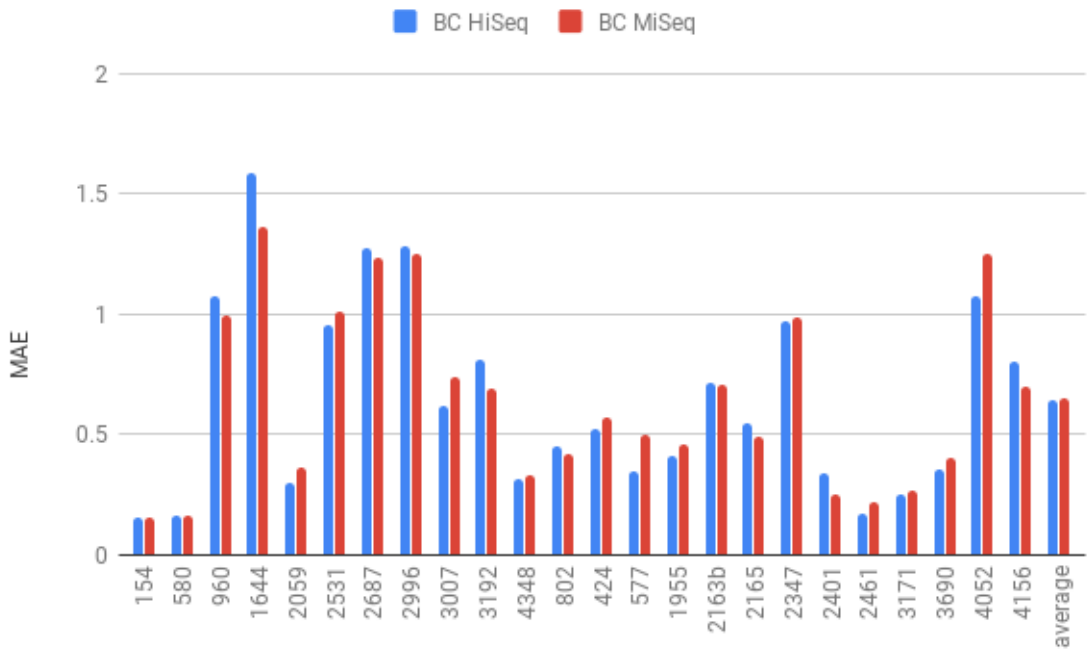


Figure B.1: MiSeq vs HiSeq trained PRINCE on BC Dataset

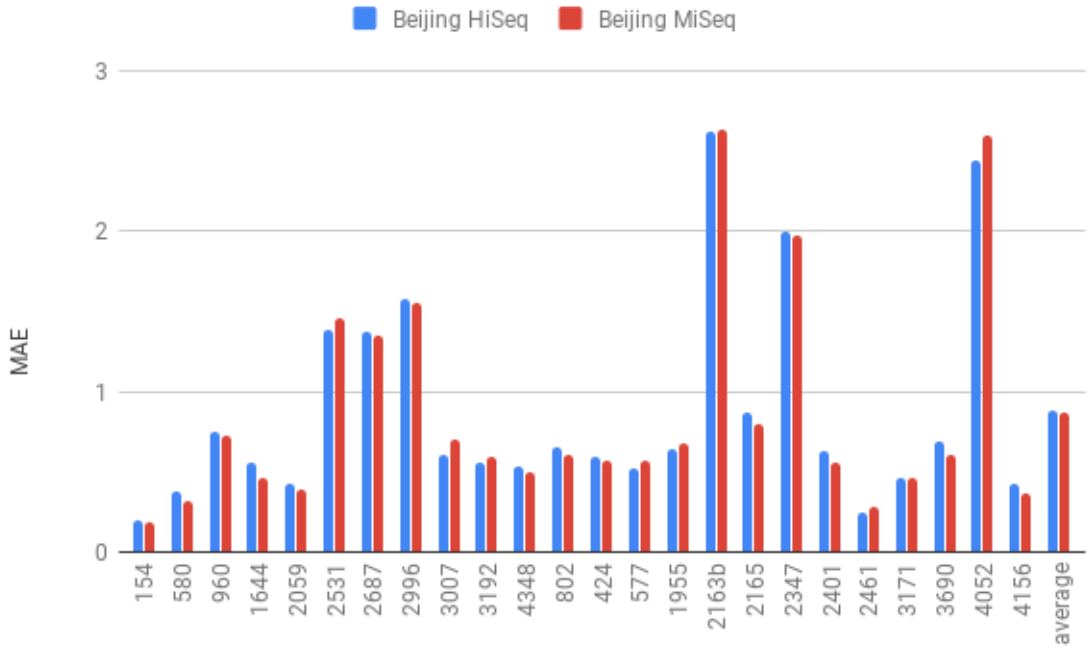


Figure B.2: MiSeq vs HiSeq trained PRINCE on Beijing Dataset

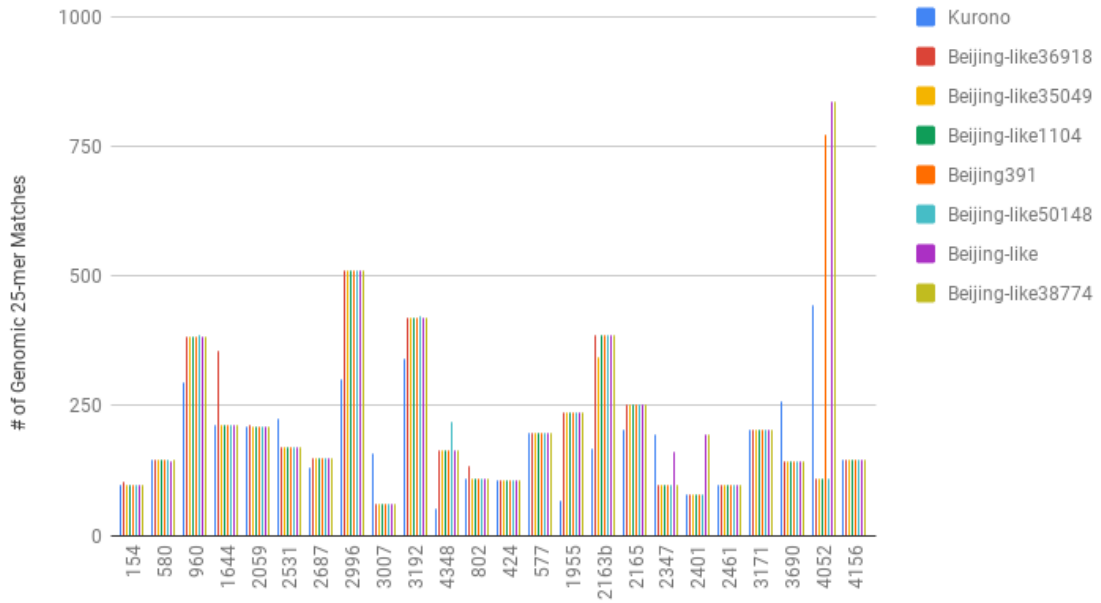


Figure B.3: Number of Recruited 25-Mers for each Genome at each VNTR

Template	Beijing PRINCE	BC PRINCE	Beijing ExpansionHunter	Kelowna ExpansionHunter	Beijing adVNTR	Kelowna adVNTR
154	0.20	0.15	4.59	0.33	0.82	1.00
580	0.38	0.16	0.93	0.00	0.39	0.21
960	0.75	1.07	0.90	1.73	0.46	1.83
1644	0.55	1.59	0.00	0.00	0.72	0.42
2059	0.42	0.30	1.58	0.92	0.38	0.21
2531	1.38	0.95	0.93	0.88	2.10	2.00
2687	1.38	1.28	4.76	4.79	0.95	1.00
2996	1.58	1.28	2.66	3.21	3.40	2.42
3007	0.61	0.62	2.30	0.85	0.68	0.00
3192	0.55	0.82	1.30	0.88	2.07	0.83
4348	0.53	0.31	1.57	0.73	1.32	0.67
802	0.66	0.45	1.95	2.40	0.84	0.21
424	0.60	0.53	2.05	1.31	1.60	1.00
577	0.52	0.35	0.94	1.56	1.04	0.79
1955	0.64	0.41	0.85	0.80	3.48	0.79
2163b	2.62	0.71	3.70	2.69	2.78	1.04
2165	0.87	0.55	2.31	4.25	2.39	1.83
2347	2.00	0.97	1.25	1.56	1.24	1.00
2401	0.63	0.34	1.35	0.57	2.35	2.17
2461	0.25	0.17	2.97	1.38	1.03	1.04
3171	0.46	0.25	1.69	0.35	0.49	0.00
3690	0.69	0.35	1.19	0.56	0.73	0.00
4052	2.44	1.08	2.26	2.17	5.70	5.21
4156	0.42	0.80	2.55	1.52	0.89	0.79

Table B.1: Template MAEs for each Method

BioProject	BioSample	Strain
PRJNA269967	SAMN05833364	Beijing/391
PRJNA269968	SAMN05833359	Beijing-like/1104
PRJNA269969	SAMN05833363	Beijing-like/50148
PRJNA269970	SAMN05833361	Beijing-like/36918
PRJNA269971	SAMN05833362	Beijing-like/38774
PRJNA269972	SAMN03258416	Beijing-like
PRJNA224116	SAMD00000334	Kurono

Table B.2: Reference Genome Accession Numbers

ID	Sequence
154	GAGCGGCGCCTATGACTCGCGCCGGCGACGATGCACAGCGAAGCGATGAGGAG
580	GATGACCTGCGCCGACGACGATGCAGAGCGTAGCGATGAGGTGGGGGCACCACCCGCTTGCGGGGGGAGAGTGGCGCT
960	AGTCAATGGCGCCGCTCCTCCTCATCGCTGCGCTCTGCATCGTCGCCGGCGGT
1644	GGCGCGGTTCACGAGCGCCGCTCCTCCTCATCGCTTCGCTCTGCATCGTCGTC
2059	GAGCGGCGCCAATGAGCCGCGCCGGCGACGATGCAGTGGGGGTACCGCCCGCTTGCGGGGGACGAAGCGATGACGAG
2531	CTCATCTGCGCCGCTCCTGCTCATCGCTTCGCTCTGCATCGTCACCGGCGCGA
2687	GTGACCGCGCCGGCGACGATGCAGAGCGCAGCGATGCTGAGGAGCGGCGCCA
2996	TGACCTCCGCCGGCGACGATGCAGAGCGCAGCGATGAGGAGGAGCGGCGCT
3007	CTGCATCGTCGCCGGCGCGGGTCACTGGCGCCGCTCCTCCCCATCGCTTTGCT
3192	CATCTGCGCCGCTCCTCCTCATCGCTGCGCTCTGCATCGTCGCCGGCGCCAAC
4348	CCCGCGCCGGCGACGATGCAAAGCGCAGCGATGAGGAGGAGCGGCGCCAACGG
802	CCGGCCTAGCCCGGCGACGATGCAGAGCGAAGCGATGAGGAGGAGCAGGGCAAT
424	GCCGGTGAGCGAGCGCGCGTAGCGGGGGAGCGAACGGCGCAGTTGGCACCA
577	GCCCGACGACGATGCAGAGCGCGCAGCGCGATGAGAAGGAGTTGGGCGGTTAGGTCGA
1955	GTCTGATTGCCCGGTTCCTCCTCGCGCCGCAAACGGCGCGCATCGTCACCGGGCGTC
2163b	ATCGGACCGACACCACCCAGCGCGTTCAGGCTCAACGGAATACCAGGAATAGTAATATCAGGCACCACA
2165	CCGTTGGGTATTGATGTGGGCGGAGGGTTAGGCCCGTTTACGGTGTTCCCGGATGGCTACTCACTTCCGGCAATC
2347	GTGGCGACCCGCCGCACCCGGCTCCGCCGCGCTTGCGATCGCCACTGGACCCAACAA
2401	TGATTGCCCGGCTCCTCCTCACCCCGTTACCCGGGGCGCATCGTCGCCGAGCTCGATT
2461	AGCCCGGCGACGACCCGGGTCGGCACGACCCGGGAAGGAACCGGGCAAATCAAGCAC
3171	GGCGATCGCAAGCGCGGCGAAGCCGGGCGCAGCGGGTCGCCACCATCGGACACA
3690	AGCTCGGCGACGATGCACGCCGCGCAGCGGCGTGAGGAGGAGCCGAGCAGTCCAACAC
4052	GTGGCGACCCGCTGCACCCGGCTCTGGGGTGATTGCCTGGCTCCTCCTCGGCCCGTTTTGCGGGCCGCATTGTCGCCAGGCGCGGGGTTTGCGATCGCCACGGGGCTGATG
4156	ACGCGGGCGACCCGCCGCACCCGGCGACAAGCGCCGAGCTTGCGATCGCCCTGAATCCA

Table B.3: VNTR Template Sequences

ID	5' Flanking Sequence	3' Flanking Sequence
154	GTCGCGATTCCCGCAGCCCATCGAATTGGACTTGCAGCAATGGACCAACT	ATTTTGAGCCGGCGTCCGAGTACGGCACGAAAGCCGCCATCGGCGTCGCC
580	GGCCTATCACGACGACGAGCGACCCGAGCAGGCGCGAGAGCCCGAACTGC	GCTGGCGTTTCTGCTGCGCAAGGAGGGCTTTGAGGCCACGGTGGTGACCG
960	GTTCTTGACCAACTGCAGTCGTCCGTAGCGGGCCATGGTGTCGACGATGT	CGCTGAGTGGCAACGCCGGCGGCATCGAGGTAGCTGATCACCAAGGTGGC
1644	TCGGTGATCGGGTCCAGTCCAAGTACCTCAATGCTGCCGGCGTCCGGGCG	GCGGCCAGCGGAATTATCGCCCACAGCAGGTACCAGGGCTGCACGACGGG
2059	CGTCGCGAGTATCTCGTTGAGCTATCTCGGAGAGATGCCCTTCGAGTTAG	TACAAGTACCTGGTCGCGGTCTCCCGAGAGCCCGATCACGACAAGATTCG
2531	ACGCTCACCAGGATCGCCAAACCGGCGCTGTCGATGGCCGCAACAAAACG	GACAAAAGGGCGAACCCGTTGAGCTGGTGAACATTGTCGTCCCGGATCCC
2687	CGACCAAGATGTGCAGGAATACATTGCCGACAACGACGTGCTGGTTAATC	TGCGGCCCGGGCTCGACGTGCGGGTCGCGTTCTGTGAGCTCAACTCGCCC
2996	GCGCCTGTGCACACGTGTCGGCGGATAGGTCTACCGTCGAAATCTGTGAC	CCCGGACGGCTCCGATGCATGGGACGGCACTATTCGCCTGGTCGCCTATG
3007	TCCGGTCGGGCGACGGGTCGATCACCTCGAAAGCCTCTGCGTGCCAGTAA	ACCTTGAGTGGCACGCTCACATCGCTGTCCTGCCACTGGCGCACCAGCTC
3192	GCGAATTGCAGTCTCGATAGCGCGCAACTGATTGGCTTCATACGGCTTTA	ATCAAGACCACGTCGGCACCGATCTCCAGCGCACGCTGCGCTGCGGTGGT
4348	CGCATCGACAAACTGGAGCCAAACCTGACATATCGAATCGCATTGCGCAC	GGATGAACCGCGCCGCCATCTACAGGATGTGTGGGGCGTAGACGTTTCCG
802	GGGTTGCTGGATGACAACGTGTACACGTTTTTGGTGCGCCCGGATTCCAA	GACGCCTGGTCGTCGCGGCGCCAGCGTATCTGATTTCGCCGAGATCACCC
424	CTTGGCCGGCATCAAGCGCATTATTGAACTGACCAGTCAGGTCGAGGCGC	GCCCGACGACGGGGTTGTGGTCGAGTTGAAGAATCCCGGGCTCTGCTGCC
577	CGAGAGTGGCAGTGGCGGTTATCTCGCGGTGAGTCGCTGCAGAACCTGCA	CCGTTCCGGACTCAGAAATTCGCGATCACAATTTGCGCGTTCAAGTCATT
1955	ATGGTCGCGGTGACGCCGACGCTAGACGTCAGATCCCAGTTGTCGTCGTC	TACAGAACCAGGCGATGTTGCCGTCCGGCCCGCGCACGAAGTTGGAGCGA
2163b	ATGTTGACCGGTATCCTGATGTTAATCGTAAGGGGGATGCGGGAAATAGG	ATGCCACCATTCACTTCGACGCTCAGTGGGATGGCGGGAATGCTGAGTGT
2165	AAATCGGTCCCATCACCTTCTTATCGGGTGGTTATAGTTTGCCGGCTATT	CCTCGGGCCGATCAACATCCCGATCGAAATCGCGGGCACCCCAGGCTTCG
2347	GCCAGCCGCCGTGCATAAACCTCGGCGTCGAATCGGCGAGAACCCATGTC	CGCAGTTCCGGGTGGACCTGATCCCGTCATACAAGGCACACCGGGTGGCT
2401	CTTGAAGCCCCGGTCTCATCTGTCACAGTCACCTTTCCTACCACTCGTAA	GCGCTAGCTTGAGGTGCTAGTGCCCTACTAATGGGCGTGGGGGTTCAAGT
2461	ATGGCCACCCGATACCGCTTCAGTATCAGGGAGCGCCGCTGCCCAAGCGA	CCGCGGTGACCACCCCGGCCAACCCGTAGCTGATCCAAGATGGCCCGTCG
3171	GGTGCGCACCTGCTCCAGATAAGCCGTCAGGCCAACCACCGGGCGACTCG	GCAGCTGAAGGCCCCGCTCGATACCGTACAACCCTCCAGCAATGAGAGCC
3690	CGGTGGAGGCGATGAACGTCTTCGACGGCACACAGTCGTCCAGTACGGCA	TGGGTACCACCCGAGCCCCCGCCGTCTAAGCTTTCCCCCGTGCCGCTCTA
4052	AATCAGGTCCGGTGCACCGCGTGTTTGACCCGAACGCTCAGCTGTCGGAT	GGGAAGGACCTGGCCGGCACGGCCGATGATCGATGAACCGGAAAAACGTG
4156	GGCCCGGGTCCACCTTGGGATTTAGCAGGCTGACCACGGATTGCTCTAGT	AACATGGACGCCGGCCTGGTGGCCGCGCTCGCCGCTCACCGCGACGCCGA

Table B.4: VNTR Flanking Sequences

224325153323444234423373	SRS2774445	SRS2774521
SRS2774518	SRS1062942	SRS2774801
234315153323441444223352	SRS2774692	SRS2774726
SRS2774727	SRS2774503	SRS2774647
223325163533245544423382	SRS2774715	SRS2774747
SRS2774500	SRS2774680	SRS2774388
228225113221343244423383	SRS2774387	SRS2774426
SRS2774536	SRS2577355	SRS2774746
225425173533524244223384	SRS2577272	SRS2577201
SRS2577413	SRS2577389	SRS2577020

Table B.5: BC Data Cluster and SRA Accession Numbers

SAMEA2533363	SAMEA2533840	SAMEA2534314	SAMEA2534935
SAMEA2533372	SAMEA2533849	SAMEA2534329	SAMEA2534942
SAMEA2533380	SAMEA2533850	SAMEA2534330	SAMEA2534985
SAMEA2533423	SAMEA2533862	SAMEA2534335	SAMEA2534992
SAMEA2533434	SAMEA2533887	SAMEA2534405	SAMEA2534993
SAMEA2533435	SAMEA2533893	SAMEA2534433	SAMEA2534994
SAMEA2533463	SAMEA2533923	SAMEA2534438	SAMEA2535016
SAMEA2533465	SAMEA2533982	SAMEA2534441	SAMEA2535025
SAMEA2533542	SAMEA2533997	SAMEA2534464	SAMEA2535029
SAMEA2533555	SAMEA2534018	SAMEA2534474	SAMEA2535113
SAMEA2533569	SAMEA2534023	SAMEA2534477	SAMEA2535125
SAMEA2533580	SAMEA2534025	SAMEA2534492	SAMEA2535126
SAMEA2533583	SAMEA2534036	SAMEA2534570	SAMEA2535134
SAMEA2533593	SAMEA2534040	SAMEA2534593	SAMEA2535142
SAMEA2533609	SAMEA2534085	SAMEA2534642	SAMEA2535152
SAMEA2533628	SAMEA2534086	SAMEA2534653	SAMEA2535166
SAMEA2533675	SAMEA2534112	SAMEA2534676	SAMEA2535180
SAMEA2533682	SAMEA2534120	SAMEA2534677	SAMEA2535188
SAMEA2533688	SAMEA2534126	SAMEA2534687	SAMEA2535236
SAMEA2533692	SAMEA2534182	SAMEA2534726	SAMEA2535248
SAMEA2533740	SAMEA2534205	SAMEA2534730	SAMEA2535265
SAMEA2533743	SAMEA2534216	SAMEA2534744	SAMEA2535268
SAMEA2533751	SAMEA2534241	SAMEA2534809	SAMEA2535280
SAMEA2533759	SAMEA2534262	SAMEA2534823	SAMEA2535290
SAMEA2533772	SAMEA2534295	SAMEA2534866	SAMEA2535298
SAMEA2533777	SAMEA2534297	SAMEA2534880	SAMEA2535307
SAMEA2533779	SAMEA2534301	SAMEA2534884	
SAMEA2533789	SAMEA2534307	SAMEA2534904	

Table B.6: Beijing Data SRA Accession Numbers