PRINCE: Accurate Approximation of the Copy Number of Tandem Repeats

Mehrdad Mansouri*
School of Computing Science, Simon Fraser University, Burnaby, BC, Canada
mansouri@sfu.ca

Julian Booth*
School of Computing Science, Simon Fraser University, Burnaby, BC, Canada
julius_booth@sfu.ca

Margaryta Vityaz*
School of Computing Science, Simon Fraser University, Burnaby, BC, Canada
rvityaz@sfu.ca

Cedric Chauve
Department of Mathematics, Simon Fraser University, Burnaby, BC, Canada
cedric_chauve@sfu.ca

Leonid Chindelevitch
School of Computing Science, Simon Fraser University , Burnaby, BC, Canada
leonid_chindelevitch@sfu.ca

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.

We propose PRINCE (Processing Reads to Infer the Number of Copies via Estimation), an algorithm that is able to accurately estimate the copy number of a VNTR given the sequence of a single repeat unit 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. Our 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. We test PRINCE on the genomes of three datasets of *Mycobacterium tuberculosis* strains and show that it is more than twice as accurate as a previous method.

An implementation of PRINCE in the Python language is freely available at `https://github.com/WGS-TB/PythonPRINCE`.

2012 ACM Subject Classification Applied computing → Molecular sequence analysis

Keywords and phrases Variable-Number Tandem Repeats, Copy number, Bacterial genomics

Digital Object Identifier 10.4230/LIPIcs.WABI.2018.20

* indicates equal contribution

Funding CC and LC are funded by NSERC Discovery grants and a CIHR/Genome Canada Bioinformatics/Computational Biology grant. LC is funded by a Sloan Foundation Fellowship.

Acknowledgements The authors would like to thank Ted Cohen and Vineet Bafna for helpful discussion and Jennifer Gardy and Jennifer Guthrie for providing assistance with the data used in our analysis.

1 Introduction

Variable number tandem repeats (VNTRs) are genomic locations where identical or highly similar sequences of DNA are repeated in tandem (i.e. with no spaces in between repeats). One reason for the importance of VNTRs in bacterial genomics is their use in the identification of related bacterial strains [2, 29]. For instance, 24 VNTR loci are used for *Mycobacterium tuberculosis* typing because of their reproducible nature and highly discriminatory typing results [30]. Related bacterial strains will typically have similar or identical copy numbers (CNs) at these loci, a feature used to identify clusters of potentially related strains in the molecular epidemiology of infectious diseases [20, 19]. In addition, the study of VNTR data has been used to glean information about bacterial lineage and pathogenicity [23].

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 [17]. Unlike the previously solved problem of reconstructing spoligotypes [5], 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 [13]. Instead, they only cover small sections of the repeats and cannot be assembled to reconstruct the entire, multi-copy, VNTR region [27], preventing the direct resolution of CNs. 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. 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, biasing this ratio toward lower values.

There are three types of tandem repeats, which differ according to the length of the sequence being repeated [31], which we refer to as the *template* for the tandem repeat. The first one is short tandem repeats (STRs) or microsatellites, whose templates range between 1 and 6 base pairs. The second one, the focus of this paper, is VNTRs or minisatellites, whose templates are typically between 10 and 100 base pairs. The third type of repeat, copy number variations (CNVs), involves large templates of size 1000 base pairs and above.

There are a number of methods that work with microsatellites. These methods 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 [15] and

HipSTR [33]. In addition, a number of tools, such as STRViper [4] and STRait Razor [34], focus on identifying repeat templates, rather than predicting the CN for a given template.

A wide range of algorithms has also been developed for CNVs. They generally fall into five categories: paired-end mapping, split read, read depth, de novo assembly of a genome, and combination of the above approaches. Read depth based methods are typically used for CNV detection, and the rest are typically used for CNV identification [36]. Read depth methods typically rely on dividing the genome into windows of at least 100 base pairs [35]. However, this cannot provide enough resolution for VNTRs, whose templates typically have a length smaller than 100 base pairs. Another issue is that these methods tend to pre-train their model on the human genome, and are not directly applicable to bacterial genomes [35, 1].

Relatively few methods have been developed to address the problem of predicting CNs of minisatellites (VNTRs) from WGS data. To the best of our knowledge, there are only two such methods that are broadly applicable: CNVeM [32] and ExpansionHunter [7]. CNVem uses a probabilistic model that utilizes the inherent uncertainty of read mapping and uses maximum likelihood to estimate locations and copy numbers. 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 used basic statistical techniques to estimate the copy number from them. Unfortunately, we were not able to successfully apply CNVeM to our data, and thus report on the results of comparing our method to ExpansionHunter in this paper.

Three other existing methods work in a similar context to ours and are worth mentioning. The first one of these methods, TGS-TB [28], was specifically developed to work for *Mycobacterium tuberculosis*, the organism which we use as the pilot application in our work. However, it requires a minimum read length of at least 300 base pairs, which exceeds the most commonly used short read technologies, and is not applicable to our data. The second one, adVNTR [3], is based on a Hidden Markov model (HMM) trained on the human genome. This method is not directly comparable to ours as it uses not only the template sequence, but also information about the region flanking each VNTR. The final one, VNTRseek [12], is designed for detecting VNTRs, not predicting the CNs of VNTRs with known templates.

In this paper we propose a method called PRINCE (Processing Reads to Infer the Number of Copies Exactly), which successfully addresses the challenges of determining CNs for VNTR in bacterial genomes. PRINCE 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 copy numbers. 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 within the reads. The recruited reads are then compared to the template to compute an overall matching score based on coverage and sequence features. 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 match score using the fitted parameters.

We have tested PRINCE on two datasets consisting of *M. tuberculosis* genome reads (135 samples in total) as well as a simulated dataset. We show that the CNs estimated by PRINCE are consistently closer to the true CNs than the ones predicted by ExpansionHunter, and that the estimation remains robust for a range of coverages, read qualities and copy numbers. PRINCE is freely available at `https://github.com/WGS-TB/PythonPRINCE`.

2 Methods

2.1 Problem Description

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), and a reference genome. In this section we define the relevant terminology.

Definition 1: A *template* is the repeat unit of a VNTR, sometimes also referred to as the pattern of a VNTR. We define T 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 *copy number* c_i is the number of times a template t_i is repeated in tandem in a genome G, possibly with errors. We define $\mathbf{c}$ 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 3: The *match score* s_i is a proxy for the copy number of a template computed by PRINCE. We define $\mathbf{s}$ as the vector of match scores corresponding to each t_i in T, and write $\mathbf{s} := [s_1, \ldots, s_m]$.

Definition 4: A *read* r_j is a subsequence of length L of a genome G, possibly corrupted by sequencing errors. We define R as the set of reads, i.e. $R := \{r_j\}_{j=1}^{n}$, where n is the number of reads.

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 Overview

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 copy number vectors. Then

$$f\colon \mathcal{R} \to S$$
$$\mathbf{g}\colon S \to C$$

A different g_i is fitted to each t_i and is dependent of the reference genome G, whereas f is a recruitment and scoring algorithm that works in the same way for all the templates and is independent of the reference genome G. The process PRINCE uses is shown in Figure 1.

2.3 Parameter Fitting and Copy Number Prediction

PRINCE's $\mathbf{g}$ function is fitted using simulated reads, generated from the given reference genome G. This is done to account for all possible copy numbers. Using a range of copy numbers at each VNTR allows PRINCE to uncover the true relationship between match score and copy number. The relationship is assumed to be linear, which follows from the assumption that the depth of coverage is approximately uniform throughout the genome.

We generated artificial genomes by removing the VNTR regions in assembled genomes and inserting varying but known numbers of copies of each template back into the genome. Simulated reads were generated from these artificial genomes using the ART software [16].

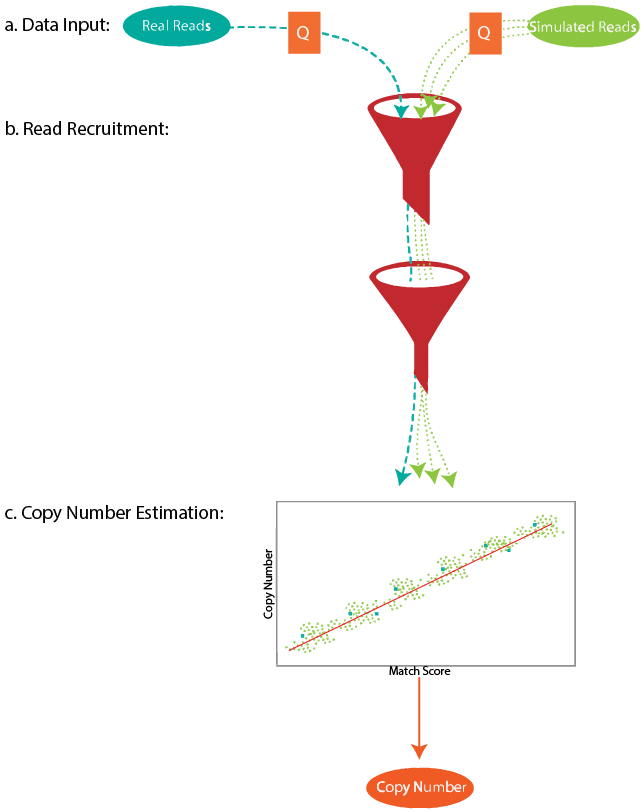


Figure 1 Flowchart of PRINCE. a. Reads are divided into mini-reads and low-quality mini-reads are discarded. b. Reads are compared against templates in two stages and match scores are computed. c. PRINCE uses the match scores to estimate copy numbers via a linear regression model.

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

$$g_i(s_i) = a_i \cdot s_i - b_i = c_i$$

for each template using a linear regression [10], and uses this function to estimate copy numbers. By fitting the coverage depth ratio 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 [8, 26]. 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 subtracted. This assumes that the erroneously recruited reads are from stable genomic regions, an assumption that appears to hold given our results.

The parameters used in PRINCE by default come from two sets of simulated reads generated from 40 artificial *Mycobacterium tuberculosis* genomes each, made from 8 real assembled genomes [25, 22] that had 1 to 5 copies of each template t_i inserted, for a total of 80 sample points. The template set T we use is the standard 24-locus MIRU-VNTR [29, 30].

2.4 Read Recruitment and Match Score Computation

In order to determine whether a read comes from a given template, we compare one to the other. This can be a computationally expensive task. To overcome this, PRINCE divides each read into its k-splits (non-overlapping k-mers), which are defined as consecutive

non-overlapping substrings of length k. Here we use $k = 25$ as most commonly used read lengths are divisible by 25. We call each of the k-splits a "mini-read". In the first step, PRINCE discards low quality mini-reads, as measured by the average Phred score [9]. We use a previously suggested minimum quality threshold of 20 [18].

PRINCE then performs a k-mer matching step. In this step all the k-mers of a mini-read, as well as its reverse complement (to allow for paired-end data), are compared to the unique k-mers of the template concatenated with itself (to allow for mini-reads that fall on the boundary between two copies) using a hash table. For this step, we use the smaller value $k = 9$, chosen via calibration. To ensure that the mini-read almost certainly comes from the VNTR region, only those reads with all k-mers matching a template k-mer (as determined via the hash table) are recruited.

The fraction of mini-reads that are recruited for template t_i out of all the ones that survive the quality filtering step gives the match score s_i. The normalization by the number of surviving mini-reads ensures that all the read sets from the same genome have the same expected match score for the same VNTR, no matter what the read length or depth of coverage is.

2.5 Simulated Data Generation

We generated a simulated dataset for testing PRINCE. 18 sets of reads were generated from 3 fully assembled genomes (Beijing-391, Beijing-like-1104, Beijing-like-35049) [25] of 6 different kinds using ART: HiSeq 2500 profile with read length of 50bp, 100bp and 150bp, HiSeqX PCR free with read length of 150 bp, and MiSeq v3 with read lengths of 200bp and 250bp.

3 Results

We have evaluated our method on three different datasets: the BC dataset (accession numbers are in Table 1 of the Supplementary Materials), the Beijing dataset, and a simulated dataset. Each one enabled us to explore the performance of our method under various conditions.

The BC dataset is a subset of the dataset collected in British Columbia from 2005 to 2014 [14], 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 Beijing dataset [21] 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 copy numbers for these datasets were established using standard experimental protocols for VNTR copy number prediction [11] 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 [6]. 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 [24].

The reason for introducing the simulated dataset is to be able to verify the method on a dataset with known copy numbers (ground truth). We have attempted to compare PRINCE to ExpansionHunter and CNVeM. Unfortunately, we were unable to use CNVeM as we could not successfully compile its source code. ExpansionHunter was designed to work for diploid organisms, and outputs two estimates and their confidence intervals. In order to correctly interpret the results, we used the median of the widest confidence interval, as suggested by the authors of ExpansionHunter.

3.1 Performance

We expected PRINCE to perform better on simulated reads than on real reads as it was trained on reads produced by ART. Indeed, PRINCE had a mean absolute error (MAE) of 0.57 over all 24 VNTR regions and 18 simulated genomes, but the MAE increased to 0.80 for both the BC and Beijing datasets, which is somewhat worse than the simulated data. The drop in performance is likely due to dissimilarities between the simulated training data and the real datasets, due to factors that may include a higher presence of read errors and varying coverage in the real data. Fortunately, PRINCE accounts for differences in input in its match score. We tested how well it adapts to variability in the input on the Beijing dataset, which had the most variation in coverage and read quality.

There was no change in PRINCE's performance across different coverages ($R^2 = 0.002$) (Fig. 3). We expect that performance might 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.

PRINCE performed well on lower quality datasets (Fig. 4). However there was a slight increase in MAE as read quality diminished ($R^2 = 0.128$). We measured dataset quality by measuring the average Phred score per nucleotide.

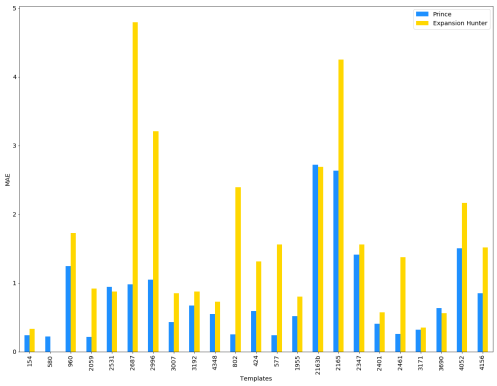
PRINCE was trained on copy numbers between 1 and 5. Some real datasets have VNTR regions with copy numbers above 10. We wanted to know how well PRINCE can extrapolate to higher copy numbers. We simulated 27 datasets using ART from genomes with copy number insertions ranging from 1 to 90 (Fig. 5). PRINCE exhibits a linear increase in MAE as copy number increases. However, PRINCE still has an MAE under 1 for copy numbers below 15. No VNTR in any genome from the two real datasets we 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, like when the number of copies increases. Although we only chose to train PRINCE with copy numbers from 1 to 5, PRINCE still performed well on the copy numbers seen in our testing datasets. PRINCE can be trained with arbitrarily high copy numbers, as long as the length of the repeat region does not become significant relative to the genome length and affect the depth of coverage calculation. Training PRINCE with high copy number data may be necessary when estimating high copy number VNTR regions.

3.2 Comparison with ExpansionHunter

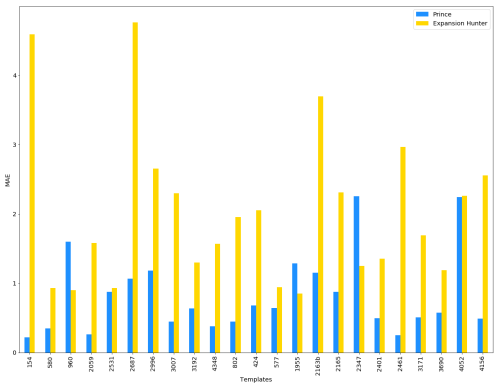
PRINCE outperforms ExpansionHunter on all three datasets. ExpansionHunter had an MAE of 1.80 for the simulated dataset and 1.62 and 2.07 for the BC and Beijing datasets respectively, an error over twice that of PRINCE in each case. ExpansionHunter performs particularly poorly on the Beijing dataset, most likely due to the variable read quality.

We performed a per-template comparison with ExpansionHunter on our three datasets. PRINCE greatly outperforms ExpansionHunter on most templates, with a few exceptions.

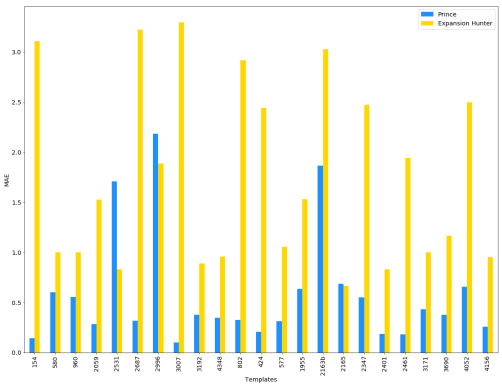
More specifically, PRINCE performs worse than ExpansionHunter on locus 2163b on the BC dataset (Fig. 2a). However, PRINCE performs relatively well on 2163b in the Beijing dataset (PRINCE : 1.15, ExpansionHunter : 3.70) (Fig. 2b). The true values for these datasets were determined using traditional typing 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 [11]. 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



(a) BC



(b) Beijing



(c) Simulated

Figure 2 MAE for PRINCE and ExpansionHunter for BC, Beijing and simulated datasets.

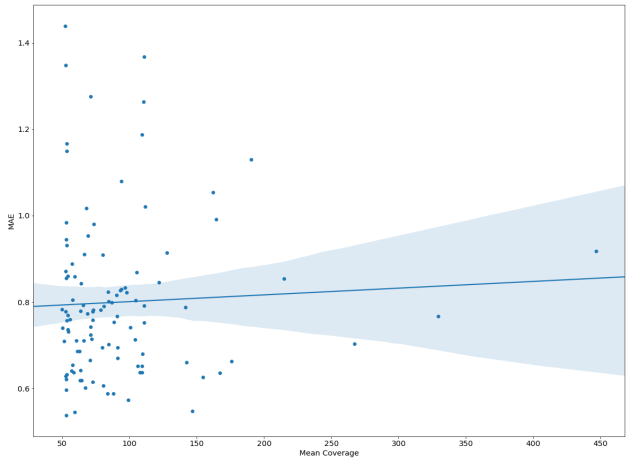


Figure 3 MAE as a function of coverage over the Beijing dataset.

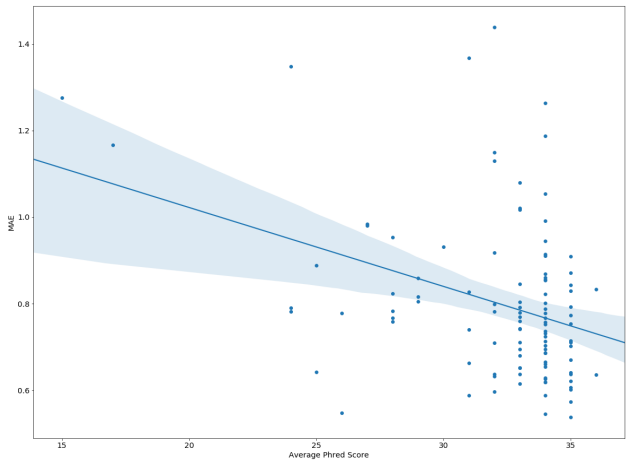


Figure 4 MAE as a function of Phred score over the Beijing dataset.

the Beijing dataset was 5.51 compared to 3.00 in the BC dataset. The poor performance of PRINCE on 2163b might be due to incorrectly labeled true values. The poor performance seen on loci 2163b, 2531 and 2996 in the simulated dataset (Fig. 2c), is more curious, as the reads come from assembled genomes. This may be due to using only three samples. Additionally, the three loci have the highest average copy number of all templates in the simulated genomes, at 8.66, 5 and 7 copies respectively. Again, training PRINCE with higher copy numbers may improve performance on these loci.

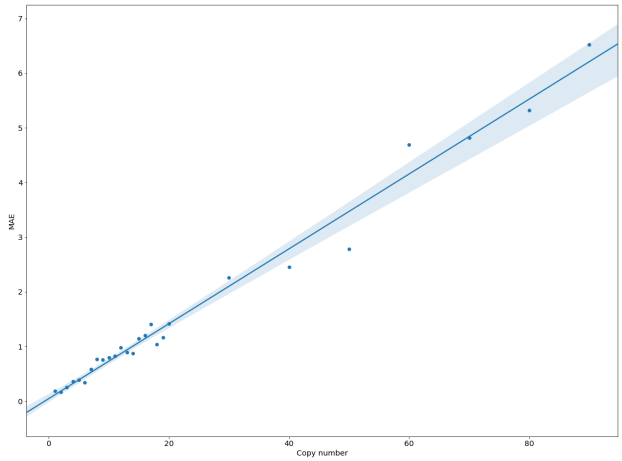


Figure 5 MAE as a function of copy number over the simulated dataset. Note that the relative error remains relatively constant over the full range of copy numbers explored.

3.3 Running Time

With Illumina paired-end WGS data at 50x coverage, PRINCE takes an average of 156 seconds per genome to query. Training takes the same amount of time per genome; however, training may use hundreds of genomes. PRINCE allows for each genome to be processed in parallel, and we were able to train PRINCE using 80 genomes in under 5 minutes.

4 Conclusion

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 loci using simulated data. We provide an example of how it could be used to estimate the copy numbers for *Mycobacterium tuberculosis* genomes, where these values could be used to compare bacteria sequenced with WGS technology to those interrogated only at their VNTR loci. In the past, this could only be done with an experimental technique specialized for tandem repeat amplification, rather than computationally from WGS data.

As input, in addition to a reference genome PRINCE only requires a set of reads and a set of templates. Unlike the method described in TGS-TB [28] that computes copy numbers exclusively for *Mycobacterium tuberculosis*, and requires reads of a minimum length of 300 base pairs and computes the copy number exclusively for predefined templates. PRINCE can be trained on any bacterial genome, can work with very short reads, and can use any set of templates.

Our work presents the first software tool that is designed to accurately infer copy numbers of VNTR templates directly from WGS data for any bacterial species. We believe that this will 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*, which we use here as an illustrative example.

References

1 A Abyzov, A E Urban, M Snyder, and M Gerstein. CNVnator: an approach to discover, genotype, and characterize typical and atypical CNVs from family and population genome sequencing. *Genome Research*, 21(6):974–984, 2011.

2 Lindstedt B. Multiple-locus variable number tandem repeats analysis for genetic fingerprinting of pathogenic bacteria. *Electrophoresis*, 26(13):2567–2582, 2005.

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

4 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.

5 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.

6 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.

7 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.

8 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.

9 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.

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

11 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.

12 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.

13 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.

14 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.

15 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.

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

17 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.

18 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.

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

20 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.

21 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.

22 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.

23 CA Nadon, E Trees, LK Ng, E Møller Nielsen, A Reimer, et al. Development and application of MLVA methods as a tool for inter-laboratory surveillance. *Euro Surveillance*, 18(35), 2013.

24 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.

25 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.

26 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.

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

28 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.

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

30 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.

31 DW Ussery, TM Wassenaar, and S Borini. *Computing for Comparative Microbial Genomics: Bioinformatics for Microbiologists*, volume 8 of *Computational Biology*. Springer, 2009.

32 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.

33 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.

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

35 S Yoon, Z Xuan, V Makarov, K Ye, and J Sebat. Sensitive and accurate detection of copy number variants using read depth of coverage. *Genome Research*, 19(9):1586–1592, 2009.

36 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.

Supplementary Materials

Table 1 SRA accession numbers for the BC data and the corresponding 24 MIRU-VNTR patterns.

SRS2774518	ERS1062942	SRS2774801	SRS2774445	SRS2774521	224325153323444234423373
SRS2774727	SRS2774503	SRS2774647	SRS2774692	SRS2774726	234315153323441444223352
SRS2774500	SRS2774680	SRS2774388	SRS2774715	SRS2774747	223325163533245544423382
SRS2774536	SRS2577355	SRS2774746	SRS2774387	SRS2774426	228225113221343244423383
SRS2577413	SRS2577389	SRS2577020	SRS2577272	SRS2577201	225425173533524244223384