Chapter 2

Bacterial genome-wide association studies

Declaration of contributions

Jukka Corander, Stephen Bentley and Julian Parkhill supervised this work. The fsm-lite k-mer counting software was written by Niko Välimäki. The initial generation of p-values of k-mers associated with antibiotic resistance using sequence element enrichment analysis (SEER) was performed by Minna Vehkala. Mark Davies, Andrew Steer and Stephen Tong collected the *S. pyogenes* isolates. I performed all other analyses, design, coding and maintenance of SEER, and generated all the figures.

Publication

The following has been published as:
Lees J. A., Vehkala M., Välimäki N., Harris S. R., Chewapreecha C., Croucher N. J., Pekka M., Davies M. R., Steer A. C., Tong S. Y. C., Honkela A., Parkhill J., Bentley S. D., Corander J. Sequence element enrichment analysis to determine the genetic basis of bacterial phenotypes. *Nat Commun*. 2016; 7, 12797. http://dx.doi.org/10.1101/038463

and prepared for publication as:
Lees J. A., Parkhill J., Harris S. R., Bentley SD. An evaluation of phylogenetic reconstruction using bacterial whole genomes.

2.1 Introduction

The goal of GWAS is to determine which genetic variants, anywhere in the genome, are associated with a trait of interest. For a binary phenotype, DNA from unrelated cases and controls are collected (ideally in the ratio 1:1 to maximise power). The simplicity of sample collection and the power of the resulting test has made GWAS a compelling study design in human genetics. In this I present work I undertook to apply this study to populations of bacterial genomes.

I wished to overcome the following issues, which were yet to be simultaneously solved by existing methods:

- Account for strong clonal population structure.
- A test which works for both complex and Mendelian-like traits.
- Test variation in the entire pan-genome.
- A computationally tractable method, implemented in a form others can use.

The first issue requires the development of an appropriate association test. The simplest test between a variant and binary phenotype is a χ^2 test based on the difference between observed and expected counts in a 2x2 contingency table comparing the proportion of case isolates an element is present in to the proportion of control isolates an element is present in. This does not account for population structure described in section 2.3, leading to many non-causal lineage associated variants reaching significance. Chewapreecha, Marttinen et al. (2014) showed that performing this test separately in each discrete defined population cluster, then combining the results (i.e. the CMH test) can mitigate this problem.

However, the definition of these clusters requires a core genome alignment and running external software (BAPS). The former may not always be available, and the latter can be computationally prohibitive to run. Additionally, when there are many population clusters compared to the total number of samples, power may be reduced. I first investigated the accuracy and computational requirements of a number of methods which represent bacterial population structure, with the goal of finding one which is fast to run and does not require a core genome alignment. Given such a definition of population structure, this could then included as fixed effects in a logistic regression. This is similar to a χ^2 test, but allows covariates to be included in the model fit, in this case to account for clonal population structure. I additionally gave consideration to the performance of this test when a single highly penetrant variant causes the phenotype, as for many antibiotic resistance determinants. This is closer to a Mendelian-like trait, as opposed to a complex trait which is affected by many lower penetrance variants.

The issue of assaying variation in the bacterial pan-genome relates to what variant is used as the predictor in these tests. Taking SNPs in the core genome, as in early human

GWAS, will miss phenotypes caused by diverse forms of variation. This can include indels, recombinations, variable promoter architecture, and differences in gene content as well as capturing these variations in regions not present in all genomes. I compared calling variation in terms of SNPs and COGs with k-mers – short words of DNA of length k, that have the potential to capture all these forms of variation. In the present chapter only common ($\geq 1\%$ MAF) variants are considered. The testing of rare variants ($< 1\%$) is underpowered in the sample sizes used here. The use of burden testing to approach this issue is discussed and performed in section 4.4.

Finally, after coming up with a test framework to overcome these issues, I designed the software package SEER to implement it. I used object oriented C++ code for speed and maintainability, as well as access to efficient linear algebra and optimisation packages (Sanderson, 2010; Sanderson & Curtin, 2016; D. E. King, 2009). I released SEER on github (https://github.com/johnlees/seer), where user comments have contributed to continued improvement and maintenance of the software.

The following sections describe how I dealt with each of these issues in turn. Section 2.6 then describes how the finished method was then applied to three datasets: on simulated data to compare its performance to existing methods, and two real datasets. The first real dataset tested whether known associations with antibiotic resistance can be recapitulated, and the second attempted to find new associations with virulence.

2.2 K-mers as a generalised variant

K-mers have the potential to allow simultaneous discovery of both short genetic variants and entire genes associated with a phenotype. Longer k-mers provide higher specificity but less sensitivity than shorter k-mers (Ondov et al., 2016). Rather than arbitrarily selecting a length prior to analysis or having to count k-mers at multiple lengths and combine the results, I wished to count all k-mers at lengths over nine bases long (as below this mapping specificity is poor).

Over all N samples, all k-mers over 9 bases long that occur in more than one sample are counted. All non-informative k-mers are omitted from the output; a k-mer X is not informative if any one base extension to the left (aX) or right (Xa) has exactly the same frequency support vector as X. The frequency support vector has N entries, each being the number of occurrences of k-mer X in each sample. Further filtering conditions are explained in section 2.2.1 below.

I used three different methods to count informative k-mers from all samples in a study. For very large studies, or for counting directly from reads rather than assemblies, I used an implementation of distributed string mining (DSM) (Välimäki & Puglisi, 2012; Seth et al., 2014) which limits maximum memory usage per core, but requires a large cluster to run.

DSM parallelises to as much as one sample per core, and either 16 or 64 master server processes. DSM includes an optional entropy-filtering setting that filters the output k-mers based on both number of samples present and frequency distribution. On 3 069 simulated genomes this took 2 hrs 38 min on 16 cores, and used 1Gb RAM per core. The distributed approach is applicable up to terabytes of short-read data (Seth et al., 2014), but requires a cluster environment to run.

For data sets up to around 5 000 sample assemblies (gigabyte-scale data) we implemented a single core version, fsm-lite, which is easier to install and run. We based fsm-lite on a succinct data structure library (Gog et al., 2014) to produce the same output as DSM. On 675 *S. pyogenes* genomes this took 3hrs 44min and used 22.3Gb RAM.

For comparison with older datasets, or where resources do not allow the storage of the entire k-mer index in memory, I used DSK (Rizk et al., 2013) to count a single k-mer length in each sample individually, then combined the results. I wrote the program combineKmers using an associative array in C++ to combine the results from DSK in memory. I concatenated results from k-mer lengths of 21, 31 and 41, as in Sheppard et al. (2013). This could in future be scaled to larger genome numbers by instead using external sorting to avoid storing the entire array in memory.

To get an idea of how much of the total genomic variance of the population each type of variant (gene, SNP or word) captured, I compared the site frequency spectrum (SFS) of informative k-mers with COGs and SNPs. Figure 2.1 shows this comparison for the 1 144 *S. pneumoniae* genomes described in chapter 4. The k-mer SFS is a similar distribution to the SNP SFS, though there are in total two orders of magnitude more words. There are also more fixed k-mers ($> 99\%$ allele frequency (AF)) – these are due to the core COGs seen in the final row. Removing rare variants which are not tested for association, the k-mer SFS remains representative of the two other variation types, and appears to be capturing both.

2.2.1 Filtering k-mers

Before testing for association, I filtered k-mers based on their frequency and unadjusted p-value. This reduced false positives from testing underpowered k-mers and reduce computational time. If not biologically plausible, k-mers with negative effect sizes are filtered at this point.

K-mers are filtered if either they appear in $< 1\%$ or $> 99\%$ of samples, or are over 100 bases long when counted by DSM. I also first test if the p-value of association in a simple χ^2 test (with 1 d.f.) is less than 10^{-5}, and remove it otherwise. In the case of a continuous phenotype a two-sample t-test is used instead. The effect of these filters is discussed in section 2.4.1.

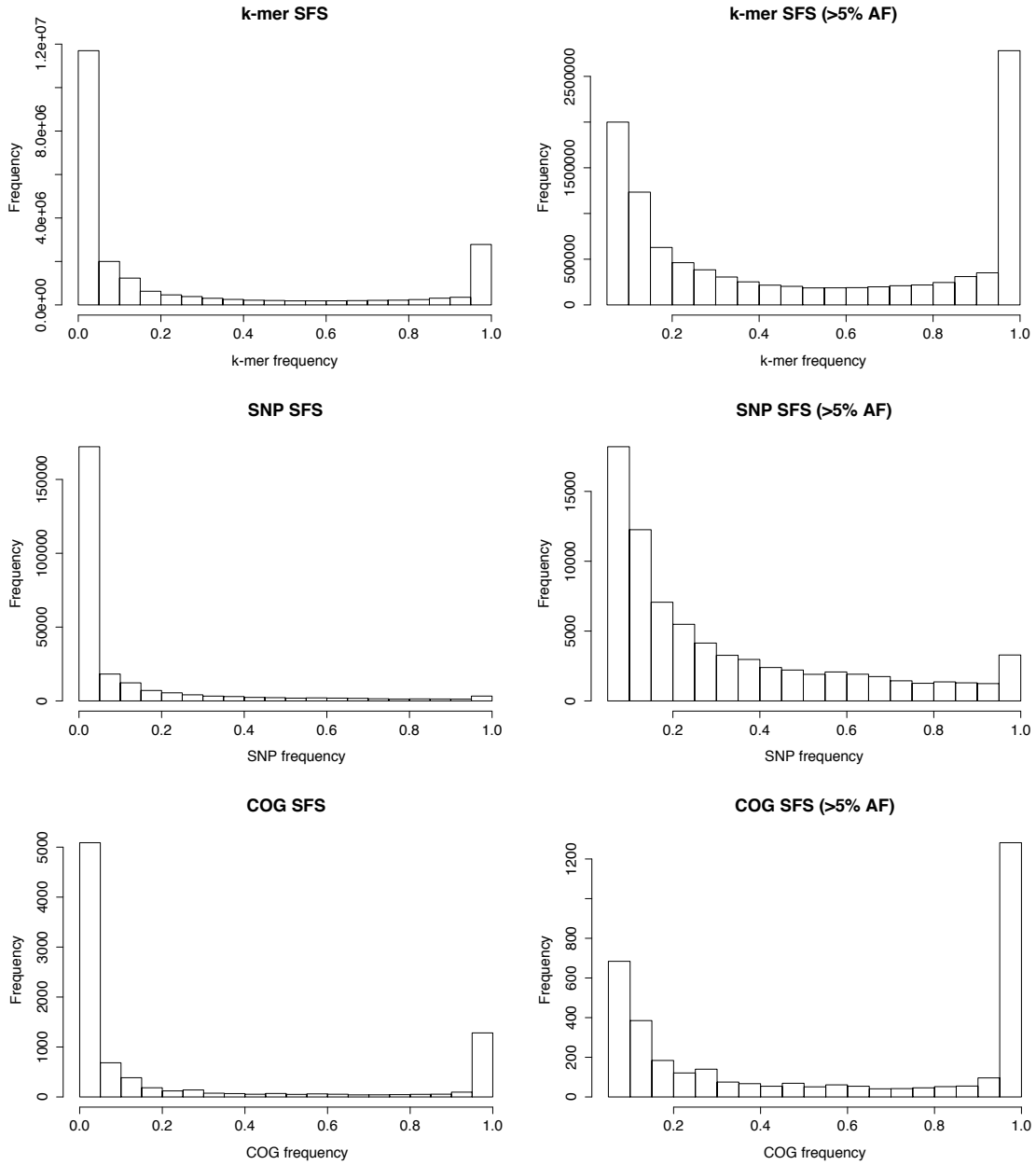


Figure 2.1: The SFS of 1 144 *S. pneumoniae* genomes. The x-axis is AF, the y-axis is the number of variants with allele-frequencies in that bin. Each row uses different sites: the first row shows k-mer presence, the second row SNPs as the sites (with respect to the ATCC 700669 reference), the third COGs. The first column shows all sites, the second column only common sites with $> 5\%$ AF.

2.3 Accounting for population structure

Due to the clonal reproduction of bacteria, rather than eukaryotic sexual reproduction resulting in recombination every generation, the genomes from a sampled population will usually be highly related. This leads to extensive LD across the chromosome, and a simple GWAS will therefore find many variants reaching significance due to their correlation with causal variants. The relatedness between all the bacteria in the study must therefore be quantified, and then appropriately used in the association model to control for this effect.

In this section I detail ways in which the population structure may be quantified, then in section 2.4 I explain how this is incorporated into an appropriate association test.

2.3.1 Phylogenetic simulation of genomes

To test the accuracy of population structure estimation, I simulated realistic data with a known phylogenetic relationship. I then used a suite of methods that infer this phylogeny from the resulting genome sequence assemblies or alignments, and evaluated them in terms of accuracy, efficiency and ease of implementation. The use of simulated data under a realistic model was desirable, as using a tree inferred from real read data as the true tree would be circular, and would necessarily result in the model that was used to infer the tree in the first place as being the most accurate.

I used artificial life framework (ALF) (Dalquen et al., 2012) to simulate evolution along a given phylogenetic tree, using the 2 232 coding sequences in the ATCC 700669 genome as the most recent common ancestor (MRCA). I used a phylogeny (fig. 2.2), originally produced by Kremer et al. (2017) from a core genome alignment of 96 *L. monocytogenes* genomes from patients with bacterial meningitis, possessing a number of qualities I wished to be able to reproduce: two distinct lineages, several clonal groups within each lineage, long branches and a polyphyletic cluster. I define N as the number of strains in the study and M as the number of aligned sites.

To estimate rates in the generalised time reversible (GTR) matrix and the size distribution of insertions and deletions, I aligned *S. pneumoniae* strains R6 (AE007317), 19F (CP000921) and *S. mitis* B6 (FN568063.) using Progressive Cactus (Paten et al., 2011). I used previously determined parameters for the rate of codon evolution (Kosiol et al., 2007), relative rate of SNPs to indels in coding regions (J. Q. Chen et al., 2009), rates of gene loss and horizontal gene transfer (Chewapreecha, Harris et al., 2014) when running the simulation. In parallel, I used DAWG (Cartwright, 2005) to simulate evolution of intergenic regions using the same GTR matrix parameters and previously estimated intergenic SNP to indel rate (J. Q. Chen et al., 2009). I combined the resulting sequences of coding and non-coding regions at tips of the phylogeny while accounting for gene loss and transfer, and finally generated error prone Illumina reads from these sequences using pIRS (Hu et al., 2012).

To generate input to phylogenetic inference algorithms, I created assemblies and alignments from the simulated reads. I assembled the simulated reads into contigs with velvet (Zerbino & Birney, 2008), then improved and annotated the resulting scaffolds (Page et al., 2016). I generated alignments by mapping reads to the TIGR4 reference using bwa-mem with default settings (H. Li, 2013), and called variants from these alignments using samtools mpileup and bcftools call (H. Li, 2011). I used Roary (Page et al., 2015) with a 95% BLAST ID cutoff to construct a pan-genome from the annotated assemblies,

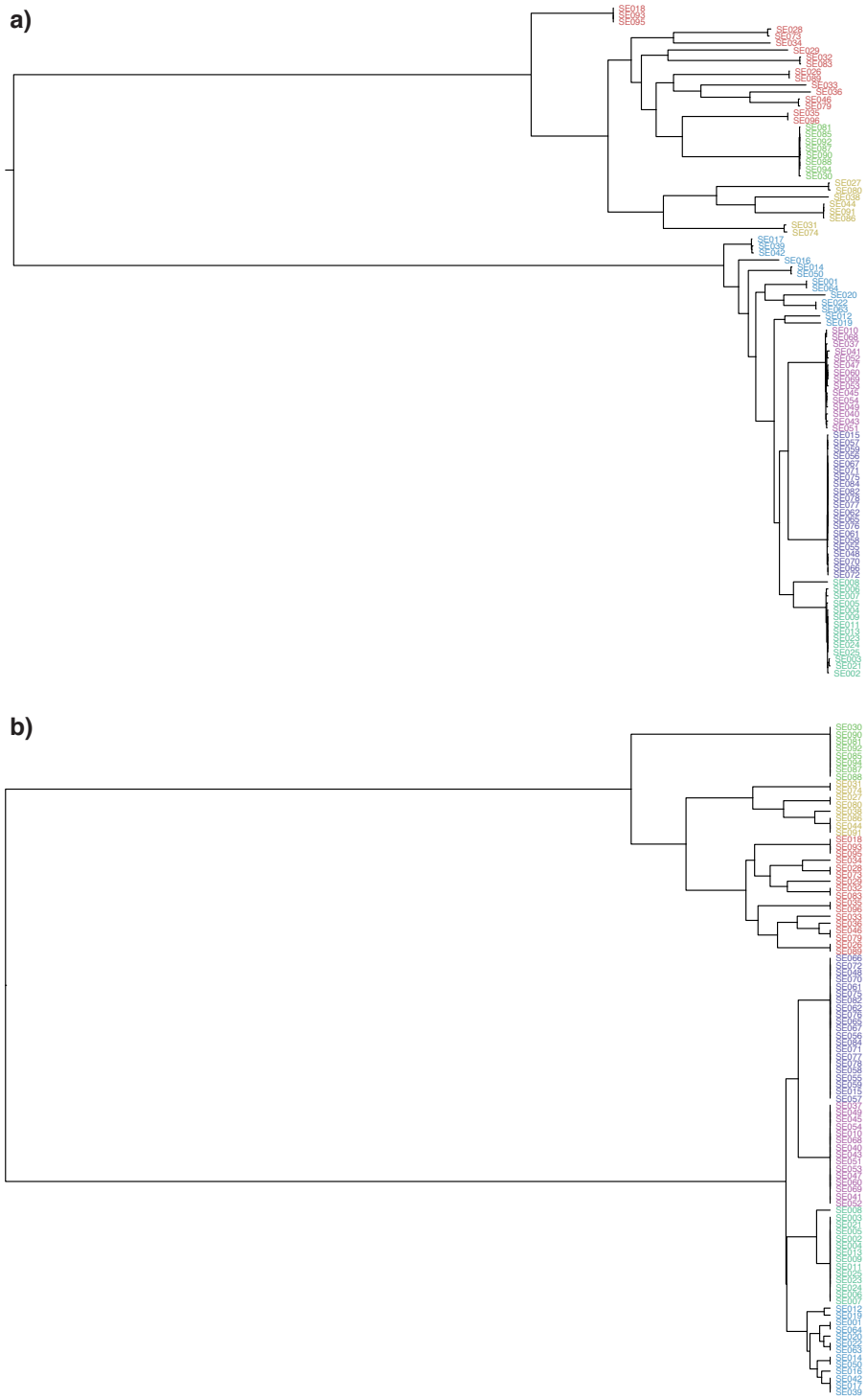


Figure 2.2: a) The phylogeny inferred by Kremer et al. (2017) used as the true tree in simulations. Tips are coloured by BAPS cluster inferred from the core genome alignment. **b)** The UPGMA tree using k-mer distances as used by SEER; tip colours are the original BAPS clusters shown in a).

from which a core gene alignment was extracted. I then created alignments by two further methods. For a MLST alignment I selected seven genes at random from the core alignment (present in all strains) which had not been involved in horizontal transfer events. For a Progressive Cactus alignment, I ran the software on the assemblies using default settings, and extracted regions aligned between all genomes from the hierarchical alignment file

and concatenated them.

Using the nucleotide alignments described above as input, I ran the following phylogenetic inference methods:

- RAxML 7.8.6 (Stamatakis, 2014) with a GTR+gamma model (-m GTRGAMMA).
- RAxML 7.8.6 with a binary+gamma sites model (-m BINGAMMA).
- FastTree 2.1.9 (M. N. Price et al., 2009) using the GTR model (denoted slow) and using the -pseudo and -fastest options (denoted fast).
- Parsnp 1.2 (Treangen et al., 2014) on all assemblies using the -c and -x options (removing recombination with PhiPack).

I also created pairwise distance matrices using:

- Mash 1.0 (Ondov et al., 2016) (default settings) between assemblies.
- Andi 0.9.2 (Haubold et al., 2015) (default settings) between assemblies.
- Hamming distance between informative k-mers using a subsample of 1% of counted k-mers from assemblies.
- Hamming distance between rows of the gene presence/absence matrix produced by Roary (using 95% blast ID cutoff).
- Jukes-Cantor (JC) and logdet distances between sequences in the alignment, as implemented in SeaView 4.0 (Gouy et al., 2010).
- Distances between core gene alleles (add a distance of zero for each core gene with identical sequence, add a distance of one if non-identical), as used in the BIGSdb genome comparator module (Jolley & Maiden, 2010).
- Normalised compression distance (NCD) (Vitányi et al., 2009), using PPMZ as the compression tool (Alfonseca et al., 2005).

For all the above distance matrix methods I then constructed a neighbour joining (NJ) tree, a BIONJ tree (Gascuel, 1997) using the R package ape, and an UPGMA tree using the R package phangorn. In the comparison I retained the tree building method from these three with the lowest Kendall-Colijn (KC) distance from the true tree.

To measure the differences in topology between the produced trees (either between the true tree and an inferred tree, or between all different inferred trees) I used two measures. As a sensitive measure of changes in topology I used the metric proposed by Kendall and Colijn (2016) with $\lambda = 0$ (ignoring branch length differences). I compared the true tree

against midpoint rooted random trees giving 286 (95% CI 276-293) as an upper limit on poor topology inference.

For trees distant from the true tree by the KC metric it was useful to test whether the tree was accurate overall and only a few clade structures were poorly resolved, or whether the tree failed to capture important clusters at all. I therefore used a measure of the clustering of the BAPS clusters from the true alignment on each inferred tree. For each pair of isolates in a BAPS cluster, a one is added to the score if any children of their most recent common ancestor is from a different cluster. I applied this to both the primary BAPS cluster, which separates the two main lineages, and the secondary BAPS clusters which define finer structure in the data. For the primary BAPS cluster a score of 0 was achieved by the true tree, which maintained these clusters, and 2437 (95% CI 2401-2457) for random trees. For the secondary BAPS clusters (excluding the 'bin' cluster) a score of 63 was achieved by the true tree, as one cluster is polyphyletic (removing this cluster gives a score of 0 to the true tree), and 535 to random trees (95% CI 531-539).

Method	KC	BAPS 1	BAPS 2	CPU time	Memory	Overheads	Parallelisability	Accessory genome?
	(0-286)	(0-2437)	(0-535)					
RAxML + close reference alignment	4.63	0	63	806.5 minutes	2.7 Gb	Mapped alignment	Pthreads	No
RAxML + alignment	11.2	0	63	587 minutes	3 Gb	Mapped alignment	Pthreads	No
Parsnp	14.0	0	63	42.5 minutes	2.6 Gb	Assemblies	Threads	No
FastTree + alignment	16.0	0	63	189 minutes	10.6 Gb	Mapped alignment	Threads (up to 4)	No
RAxML + core gene alignment	18.6	0	63	29.2 minutes	0.15 Gb	Core gene alignment	Pthreads	No
NJ + SNP alignment	20.5	0	63	Negligible	Negligible	Mapped alignment	No	No
BIONJ + mash distances	51.7	0	63	0.75 minutes	10 Mb	Assembly	Embarrassingly	Yes
RAxML + MLST alignment	62.6	0	63	1.4 minutes	19 Mb	Assembly	Pthreads	No
BIONJ + andi distances	66.0	0	60	7.48 minutes	290 Mb	Assembly	Embarrassingly	Yes
RAxML + Cactus alignment	67.2	0	63	9 600 minutes	37.4 Gb	Assembly	Threads	No
RAxML + gene presence/absence	77.3	0	57	4.28 minutes	20 Mb	Core gene alignment	Threads	Yes
BIONJ + k-mer distances	89.6	0	63	37.3 minutes	180 Mb	Assembly	Threads	Yes
BIONJ + BIGSdb	149.8	0	22	0.48 minutes	Negligible	Assembly	Embarrassingly	No
UPGMA + NCD	210	0	627	1 040 minutes	Negligible	Assembly	Embarrassingly	Yes

Table 2.1: Accuracy and resource usage of phylogenetic reconstruction methods, ordered by KC metric score. The method lists the best combinations of all alignment with phylogenetic method, and distance matrices with phylogenetic methods. Three scores of accuracy of the phylogeny are shown; values in the header are the range the values can take. Parallelisability shown is that built into the software, 'embarrassingly' is when every value in a distance matrix is independent so can be parallelised up to N^2 times.

Table 2.1 and fig. 2.3 show the results of my simulations. I used these simulations to guide the population structure correction to use in SEER bearing in mind the criteria laid out above, and also for efficiency/accuracy tradeoffs when constructing phylogenies

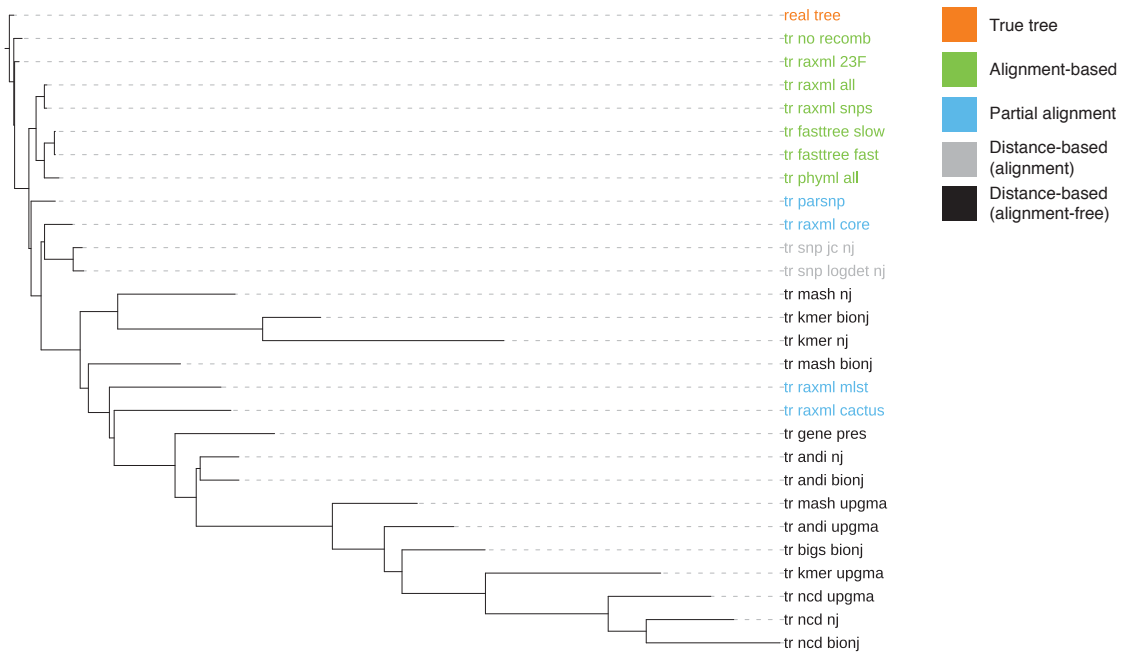


Figure 2.3: Using the KC metric between all the inferred phylogenies in table 2.1 to create a pairwise distance matrix, then an NJ tree from this matrix. This shows how the topologies from all methods are related to each other (a tree-of-trees, or supertree). The true tree is in orange and was used to root the tree, and four classes of method are labelled.

throughout the rest of this thesis.

Firstly I note that all methods except for the NCD were able to recapitulate the population clusters as defined by BAPS. Therefore for analyses which require identifying clusters on the phylogeny, but not finer scale topology, quicker but less accurate methods are sufficient. For construction of a maximum likelihood tree RAxML is currently the most efficient software available. This was the most accurate method tested, and also the most resource heavy. RAxML's model fits the way the data was generated, and is expected to be a good model of evolution. There was no significant difference in fit between the inferred tree and the true tree (likelihood ratio test (LRT) = 2.34; p = 0.13). When applied to an alignment with a reference genome more distant from the root, this method was still the most accurate. Using a core genome alignment slightly reduces the accuracy, as the number of sites M used in the inference was reduced compared to the pseudo-alignment from mapping. Using an MLST alignment of seven genes reduces the accuracy greatly, as only a small proportion of the genomic variants are now used the the inference.

I found parsnp and FastTree on a whole genome alignment to be the methods which, while slightly less accurate than RAxML, were able to produce a good quality phylogeny rapidly. This is useful for alignments with large N and M. Distance matrix and NJ methods generally performed more poorly, but were still able to resolve large scale population structure differences.

I now discuss in detail a method which fulfilled the criteria for SEER's population

structure correction: it accurately represented the BAPS clusters without needing a core-genome alignment, used only the information already needed to perform and association test on k-mers, could be efficiently implemented in C++ with the rest of SEER, and could be used to provide covariates for a logistic or linear regression rather than using discrete clusters or a phylogeny.

2.3.2 K-mer distance method producing covariates to control for population structure

Compared with modelling SNP variation, the use of k-mers as variable sequence elements has been previously shown to accurately estimate bacterial population structure (Tasoulis et al., 2014). As k-mers are going to be used as the input to the association test, it would be convenient if they could also be used to control for population strucutre. I defined the k-mer distance in table 2.1 as follows. First I take a random sample of between 0.1% and 1% of k-mers appearing in between 5-95% of isolates. I then construct a pairwise distance matrix $\boldsymbol{D}$, with each element being equal to a sum over all m sampled k-mers:

$$d_{ij} = \sum_{m} ||k_{im} - k_{jm}|| \tag{2.1}$$

where k_{im} is 1 if the mth sampled k-mer is present in sample i, and 0 otherwise. Each element d_{ij} is therefore an estimate of the number of non-shared k-mers between a pair of samples i and j, and furthermore is proportional to the Jaccard distance between the samples (Levandowsky & Winter, 1971). When I clustered samples using these distances, I got the same results as clustering core alignment SNPs using hierBAPS (L. Cheng et al., 2013) as shown in fig. 2.2b). These clusters have been used in previous bacterial GWAS studies to correct for population structure (Chewapreecha, Marttinen et al., 2014). However, this distance matrix has the clear advantage that no core gene alignment or SNP calling is needed, so it can be directly applied to the the k-mer counting result.

In an analogous way to the standard method used in human genetics of using principal components of the SNP matrix to correct for divergent ancestry (A. L. Price et al., 2006; Chengsong & Jianming, 2009), I then wrote C++ code to perform metric multidimensional scaling (MDS) on $\boldsymbol{D}$, projecting these distances into a reduced number of dimensions. The normalised eigenvectors of each dimension of this projection can then be used as covariates in the regression model, where the number of dimensions used is a user-adjustable parameter, and can be evaluated by the goodness-of-fit and the magnitude of the eigenvalues. For the tree shown in fig. 2.2, one dimension was sufficient as a population control (fig. 2.4a), whereas for the larger collection of 3 069 isolates 10-15 dimensions were needed to give tight control (fig. 2.4b). The small collection has much of the variance explained by the first dimension/eigenvector, as there is a large separation

between two main lineages. In the other collections there is a strain structure with multiple lineages, so more dimensions must be included to capture this. Over all the studies I tested, generally three dimensions appeared a good trade-off between sensitivity and specificity, but I automatically provide a scree plot as output so users can choose an appropriate number of dimensions to retain.

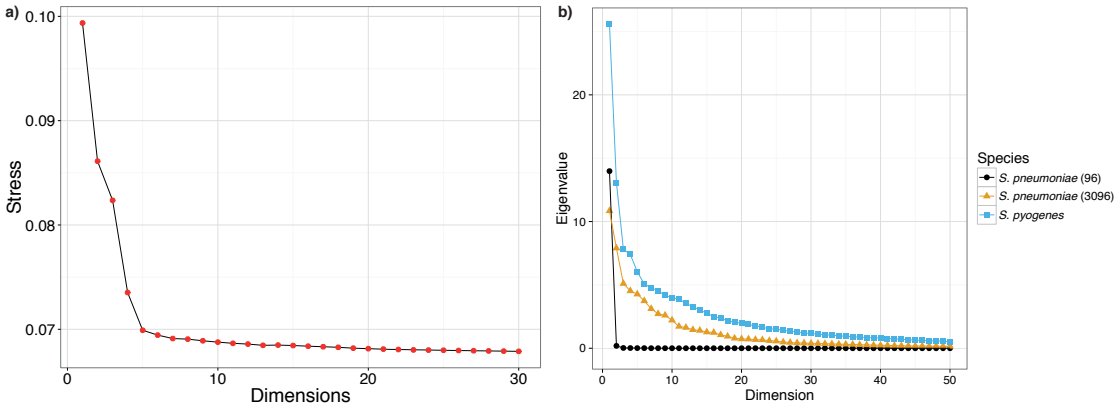


Figure 2.4: a) Stress against first thirty dimensions, calculated for the *S. pneumoniae* simulations in section 2.6.1 (orange in panel b). Stress is defined as $S^2 = 1 - R^2$, where the R^2 statistic is calculated from a regression between the upper triangle of entries in the distance matrix (i.e. pairwise between all samples) and the Euclidean distance between samples in the reduced dimension space. **b)** Eigenvalues for the first fifty dimensions of the 96 simulated *S. pneumoniae* isolates in black (section 2.3.1), 3 069 *S. pneumoniae* isolates in orange (section 2.6.1), and 675 *S. pyogenes* isolates (section 2.6.3) in blue.

I noted above that the distance used to approximate bacterial population structure is an estimate of the k-mer Jaccard distance. After the first version of SEER, the software mash was developed. This instead uses the MinHash algorithm on k-mers to estimate the Jaccard distance between sequences in a highly efficient manner (Ondov et al., 2016). As shown in table 2.1 and fig. 2.3 this distance matrix is considerably more computationally efficient than the subsampling proposed above, works from the same input data, and produced a more accurate version of the tree topology in tests. Since version `acc4bc1` I have recommended the use of mash over the above calculation I implemented in SEER, and provide scripts to run mash and MDS in a manner compatible with the rest of the package.

2.4 Association testing

Using k-mers as a generalised variant and the above population structure definition I used general linear models with fixed effects to test for association between genetic variation and phenotypes. For each k-mer, I wrote code to fit a logistic curve to binary phenotype data, and a linear model to continuous data. I took care to use time efficient optimisation routines to allow testing of all k-mers. Bacteria can be subject to extremely strong selection pressures, producing common variants with very large effect sizes, such as antibiotics

inducing resistance-conferring variants. This can make the data perfectly separable, and consequently the maximum likelihood estimate ceases to exist for the logistic model. Firth regression has been used to obtain results in these cases (Heinze & Ploner, 2003).

In detail, the SEER association testing code does the following. For samples with binary outcome vector $\boldsymbol{y}$, it fits a logistic model to each k-mer:

$$\log\left(\frac{\boldsymbol{y}}{\boldsymbol{I}-\boldsymbol{y}}\right) = \boldsymbol{X}\boldsymbol{\beta} \tag{2.2}$$

where absence and presence for each k-mer are coded as 0 and 1 respectively in column 2 of the design matrix $\boldsymbol{X}$ (column 1 is a vector of ones, giving an intercept term). Subsequent columns j of $\boldsymbol{X}$ contain the eigenvectors of the MDS projection, any input categorical covariates (automatically dummy encoded), and quantitative covariates (automatically normalised). I used the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm to maximise the log likelihood $\mathscr{L}$ in terms of the gradient vector $\boldsymbol{\beta}$ (using an analytic expression for $\mathrm{d}(\log\mathscr{L})/\mathrm{d}\boldsymbol{\beta}$):

$$\log(\mathscr{L}) \propto \sum_i [y_i \cdot \log(\mathrm{sig}(\boldsymbol{X}\boldsymbol{\beta})_i) + (1-y_i)\cdot\log(\mathrm{sig}(1-\boldsymbol{X}\boldsymbol{\beta})_i)] \tag{2.3}$$

where sig is the sigmoid function. If this fails to converge, n Newton-Raphson iterations are applied to $\boldsymbol{\beta}$:

$$\boldsymbol{\beta}_{n+1} = \boldsymbol{\beta}_n + [-\mathscr{L}''(\boldsymbol{\beta}_n)]^{-1}\cdot\mathscr{L}'(\boldsymbol{\beta}_n) \tag{2.4}$$

from a starting point using the mean phenotype as the intercept, and the root-mean squared beta from a test of k-mers passing filtering:

$$\beta_{0,0} = \frac{\sum y_i}{n}$$
$$\beta_{0,j>0} = 0.1$$

This is slower than using BFGS, but has a higher success rate.

If any entries for the observed counts in the contingency table were one or zero, or if two counts were five or less then Firth logistic regression is used instead. This regression is also used if after 1 000 Newton-Raphson iterations convergence is not reached, due to the observed points being separable, or the standard error of the slope is greater than 3 (which empirically indicated almost separable data). Firth regression adds an adjustment to $\log(\mathscr{L})$:

$$\log[\mathscr{L}(\boldsymbol{\beta})]^* = \log[\mathscr{L}(\boldsymbol{\beta})] + \frac{1}{2}\cdot\left\{\frac{d^2\mathscr{L}}{d\boldsymbol{\beta}^2}(\boldsymbol{\beta})\right\} \tag{2.5}$$

using which I applied Newton-Raphson iterations as above.

In the case of a continuous phenotype a linear model is fitted:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} \tag{2.6}$$

to find $\boldsymbol{\beta}$, I used the BFGS algorithm to minimise the squared distance $U(\boldsymbol{\beta})$:

$$U(\boldsymbol{\beta}) = ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||^2 \tag{2.7}$$

If this fails to converge then the solution is instead obtained by orthogonal decomposition of the design matrix:

$$\boldsymbol{X} = \boldsymbol{Q}\boldsymbol{R} \tag{2.8}$$

then back-solving for beta in:

$$\boldsymbol{R}\boldsymbol{\beta} = \boldsymbol{Q}^T\boldsymbol{y} \tag{2.9}$$

For both the logistic and linear model the standard error on the slope β_1 is calculated by inverting the Fisher information matrix $\mathrm{d}^2\mathscr{L}/d\boldsymbol{\beta}^2$ to obtain the variance-covariance matrix. Inversions are performed using the Cholesky decomposition, or if this fails due to the matrix being almost singular I used the Moore-Penrose pseudoinverse. In the initial version of SEER, I used the Wald statistic to test the probability null hypothesis of no association ($\beta_1 = 0$)

$$W = \frac{\beta_1}{\mathrm{SE}(\beta_1)} \tag{2.10}$$

which is the test statistic of a χ^2 distribution with 1 d.f. This is equivalent to the positive tail of a standard normal distribution, one minus the integral of which gives the p-value.

The Wald test loses power when large effect sizes are tested (Agresti, 2015); I observed this when testing k-mers of a mosaic *penA* allele which are known to be causal for cephalosporin resistance in *Neisseria gonorrhoeae* (Unemo & Shafer, 2014). A χ^2 test gave a p-value of 3.5×10^{-181} whereas a logistic regression using the Wald test gave a p-value of 1.9×10^{-45}, less significant than some non-causal k-mers. A better test is the LRT: in this case, the LRT of the logistic model gave a p-value of 8.4×10^{-190}, making these k-mers the top hit.

Here, the LRT test statistic D is defined as

$$\begin{aligned} D &= -2 \cdot \log\left(\frac{\mathscr{L}(\text{alternative model})}{\mathscr{L}(\text{null model})}\right) \\ &= 2 \cdot [\log\{\mathscr{L}(\beta_1 = \beta_{\text{fit}})\} - \log\{\mathscr{L}(\beta_1 = 0)\}] \end{aligned}$$

using eq. (2.3) as the likelihood. The distribution of D is χ^2 with $\mathrm{df}_{\text{alt}} - \mathrm{df}_{\text{null}}$. In this case, two times the difference between the log-likelihood at the fitted value and the log-likelihood of a fit where the k-mer presence/absence column is removed from the design

matrix is tested using a χ^2 distribution with one degree of freedom. Since version `038c4cd` of SEER the p-value for logistic regression is instead calculated using the LRT by default, though the Wald test p-value is still reported for backwards compatibility.

2.4.1 Significance cut-off

For the basal cut-off for significance I used $p < 0.05$, with which I used the conservative Bonferroni correction for multiple testing to give the threshold 1×10^{-8} based on every position in the *S. pneumoniae* genome having three possible mutations (Ford et al., 2013), and all this variation being uncorrelated. This is a strict cut-off level that prevents a large number of false-positives due to the extensive amount of k-mers being tested, but does not over-penalise by correcting directly on the basis of the number of k-mers counted. To calculate an empirical significance testing cut-off for the p-value under multiple correlated tests, I generated the distribution of p-values from 100 random permutations of phenotype. For the 3 069 Maela genomes setting the FWER at 0.05 gave a cut-off of 1.4×10^{-8}, supporting the above reasoning.

In general, the number of k-mers and the correlations between their frequency vectors will vary depending on the species and specific samples in the study, so the p-value cut-off should be chosen in this manner (either by considering possible variation given the genome length, or by permutation testing) for individual studies. I have also included association effect size and p-value of the MDS components in the output of SEER, to compare lineage and variant effects on the phenotype variation.

The effect the initial χ^2 filtering step can be seen by plotting the unadjusted and adjusted p-values of the k-mers from the simulated data set described in section 2.6.1 against each other (fig. 2.5). 430 k-mers of 12.7M passing frequency filtering have an unadjusted p-value which fail to meet the χ^2 significance threshold, but would be significant using the adjusted test (and have a positive direction of effect). These k-mers were all short words (10-21 bases; median 12) that appear multiple times per sample, and therefore are of low specificity. When I tested the top p-value k-mer in this set it showed a strong association of the presence/absence vector with three population structure covariates used ($p = 1.4 \times 10^{-24}$; $p = 1.2 \times 10^{-46}$; $p = 1.5 \times 10^{-9}$ respectively). Using lasso regression, the second population structure covariate has a higher effect in the model than the k-mer frequency vector. Together, this suggested that these filtered k-mers are associated to a lineage related to the phenotype, but are unlikely to be causal for the phenotype themselves. To confirm this, I mapped these k-mers back to the reference sequence. None of these k-mers map to the gene causal to the phenotype.

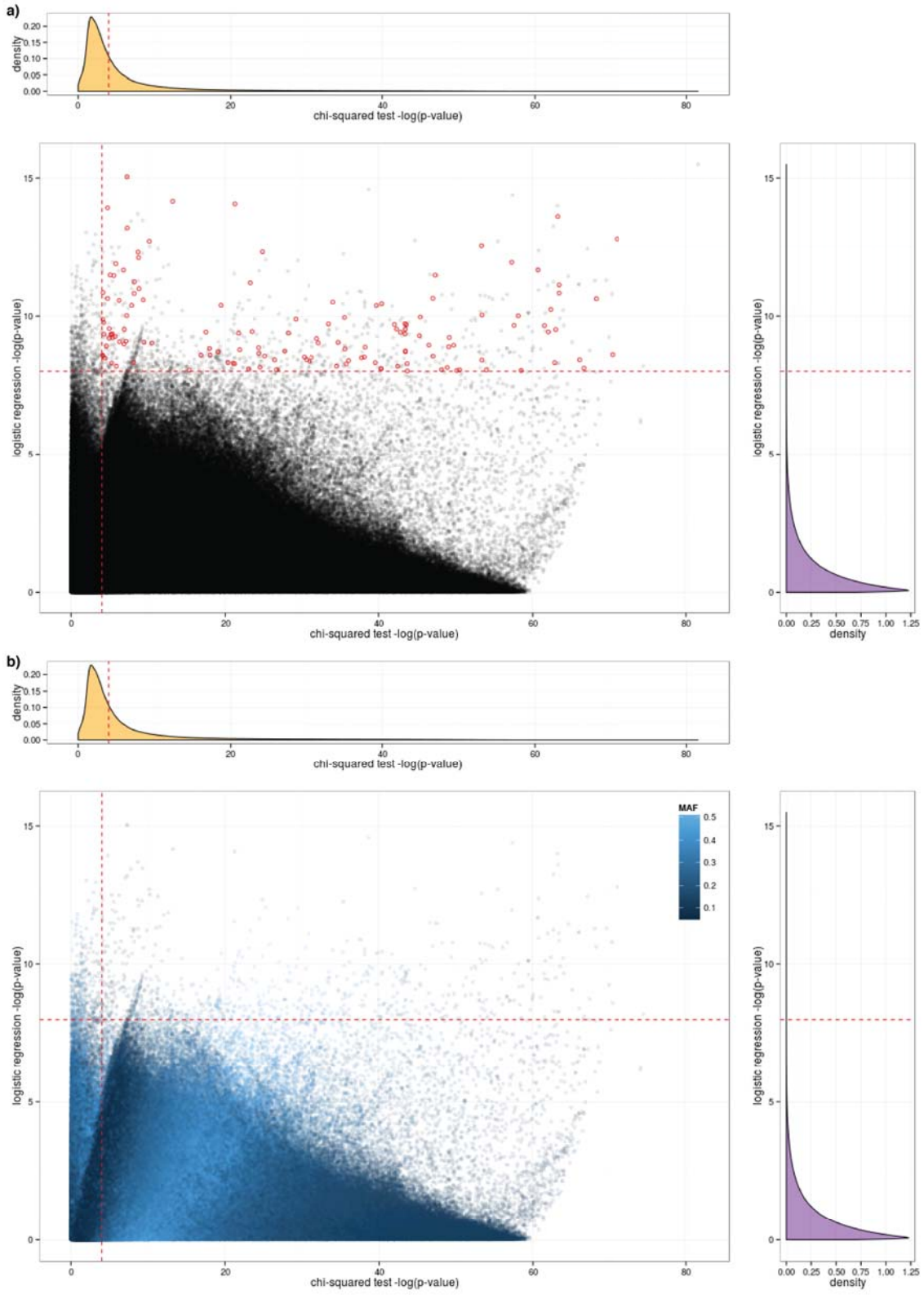


Figure 2.5: The $-\log_{10}$ p-values from a χ^2 test against the p-value from a logistic regression using the first three MDS components as covariates. The points are from all the simulated k-mers passing frequency filtering. The cut-offs used for each test are shown as red dashed lines. Top panel: marginal distribution of χ^2 p-values. Right panel: marginal distribution of logistic regression p-values. **a)** k-mers meeting the threshold for significance (a cut-off of 1×10^{-8}) in the logistic regression which map to the causal gene are coloured in red. **b)** shading of each point is by MAF. Most of the k-mers with a high χ^2 p-value and low logistic regression p-value are at low frequency, as are those with equal p-values from each test.

2.4.2 Downstream interpretation of significant k-mers

Significant k-mers can be interpreted directly through mapping to annotated genomes, or by assembling them first. Assembly may be better at searching for gene clusters associated with phenotype as longer and more specific k-mers will be generated. I assembled significant k-mers assembled using Velvet (Zerbino & Birney, 2008) choosing a smaller sub-k-mer size which maximises longest contig length of the final assembly. K-mers in the output which are substrings of other longer significant k-mers are removed.

I used BLAT (Kent, 2002) with a step size of 2 and minimum match size of 15 to find inexact but close matches to a well annotated reference sequence. Small k-mers are more likely than full reads to map equally well to multiple places in the reference genome, so reporting both mappings increases the sensitivity. For the tested dataset an average of 21% of k-mers significantly associated with antibiotic resistance report secondary mappings. These k-mers are short (median 15bp), and therefore have low specificity and high sensitivity as expected. I wrote a script which combines the p-values from SEER and co-ordinates from mapping of the significant k-mers into a .plot file, which can be loaded into visualisation software http://jameshadfield.github.io/phandango/ to create a Manhattan plot.

When k-mers do not map to a reference genome, I wrote the C++ program map_back to help interpret these. This reads in all the tested assemblies from which the k-mers were generated into memory, and threads are spawned which search for k-mers (and their reverse complement) by exact string match. Using the mapped co-ordinates, annotations of features in these regions can examined for overlap of function.

2.5 Development of SEER

I implemented SEER in C++ using the armadillo linear algebra library (Sanderson, 2010; Sanderson & Curtin, 2016), and dlib optimisation library (D. E. King, 2009). When the code was stable, I profiled its execution over a test dataset of 1 000 k-mers. Most of the processing time was spent evaluating the exp() function, which is required O(N) times per k-mer when calculating the likelihood function and its gradient during the logistic fit, where N is number of samples. I was satisfied that this demonstrated an efficient usage of CPU time, and further did not identify any memory leaks when profiling with valgrind.

For ease of deployment on non-cluster machines I also threaded each filtered k-mer's fitting routine; on four cores this achieved a 2.1 times speedup. While this could probably be improved by increasing the number of k-mers handled by each thread, the algorithm is embarrassingly parallel – in practise I split the k-mer file into 16 and ran an independent process on each one. I also threaded the calculation of entries in the distance matrix D, using mutex locks to ensure only one process wrote an entry to the matrix at a time. This

was over 99% efficient.

On my simulation of 3 069 diverse 0.4Mb genomes described in section 2.6.1, 143M k-mers were counted by DSM and 25M 31-mers by DSK. On the largest DSM set, using 16 cores and subsampling 0.3M k-mers (0.2% of the total), calculating population covariates took 6hr 42min and 8.33GB RAM. This step is $O(N^2M)$ where M is number of k-mers, but can be parallelised across up to N^2 cores.

Processing all 143M informative k-mers as described took 69min 44s and 23MB RAM on 16 cores. This step is $O(NM)$ and can be parallelised across up to M cores.

After the initial release I added the following features, fixes and improvements in response to user comments on github:

- Convergence errors and the type of regression used are added in a comment field for each k-mer.
- Created a virtual machine with SEER installed, without the requirement for further dependencies.
- Statically complied version (includes libraries in executable).
- Add scripts to map significant k-mers and create a Manhattan plot.
- An alternative implementation of the population structure correction, written in R.
- Tests of all features of SEER, and continuous integration of these through travis.
- Improved installation and usage instructions, including a self-contained tutorial.

2.6 Benchmarking SEER

I benchmarked the performance of SEER on three datasets. The first was a large simulated set of *S. pneumoniae* genomes where I was able to define the associated element and set its effect size manually – this allowed me to calculate the discovery power of SEER for different sample sizes under different situations. The second dataset was 3 069 real *S. pneumoniae* genomes with five antibiotic resistance phenotypes available which helped me evaluate whether SEER could capture both gene and SNP mediated resistances (which have large effect sizes, and are often homoplasic, so should be easy to find), and how SEER compares to previous methods. Finally, I tested SEER on 675 *S. pyogenes* genomes from invasive and non-invasive samples to see if SEER could discover any new associations with a clinically relevant phenotype other than resistance.

2.6.1 Simulated data

I used a framework similar to that described in section 2.3.1 to simulate genetic sequences. To make running the simulation tractable for such a large population size, I took a random subset of 450 genes from the *S. pneumoniae* ATCC 70066916 strain as the starting genome for ALF (Dalquen et al., 2012). Using the same parameters as in section 2.3.1 I simulated 3 069 final genomes along the phylogeny observed in a Thai refugee camp (Chewapreecha, Harris et al., 2014). pIRS (Hu et al., 2012) was again used to simulate error-prone reads from genomes at the tips of the tree, which I then assembled by Velvet (Zerbino & Birney, 2008). DSM was used to count k-mers from these de novo assemblies. I counted 143M informative k-mers from this simulated data, though on the real dataset of full length genomes only 68M informative k-mers were counted.

I used a gamma plus invariant sites model as the distribution of rate heterogeneity among sites. As I did not have estimates for the parameters of this distribution directly from the data, I used the estimate given by ALF. The resulting gamma distribution must have a longer tail than the real data, as some sites vary at high frequency. This created many low-frequency k-mers. As the simulation is computationally very expensive to run, I decided that rather than running it lots of times with different parameters until a k-mer distribution identical to the observed data was reached it would be sufficient to use the original result. The excess of low frequency k-mers would be filtered out in the common variation associations I am testing. 24.7M k-mers passed frequency filtering from the real data, whereas 12.7M passed from the simulated data – while this wasn't quite the linear scaling expected with genome length (which would predict around 7M k-mers) the amount of common variation at the gene level was similar to real data. For the purpose I used the simulations for, a gene driven association at different ORs, this result was still an appropriate test.

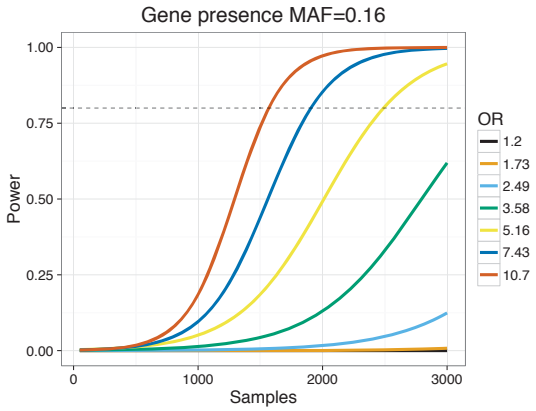
I then simulated the phenotype based on the genetic sequence. I set the ratio of cases to controls in the population (S_R) at 50% to represent typical antibiotic resistance, and designated a single variant (which could be either gene presence/absence or a SNP) as causal. MAF in the population is set from the simulation of genomes, and OR can be varied. The number of cases D_E is then the solution to a quadratic equation (Newman, 2003), which is related to probability of a sample being a case by

$$P(\text{case}|\text{major allele}) = \frac{D_E}{\text{MAF}} \tag{2.11}$$

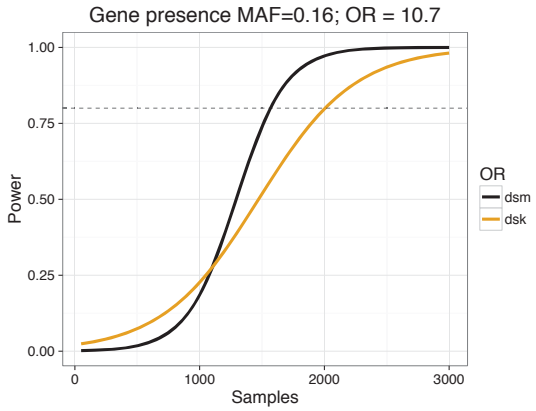
$$P(\text{case}|\text{minor allele}) = \frac{\frac{S_R}{S_R+1} - D_E}{1-\text{MAF}} \tag{2.12}$$

I generated random subsamples of the population 100 times at a range of sample sizes below the total, with case and control status assigned for each run using these formulae. I defined power by the proportion of runs that had at least one k-mer in the gene significantly

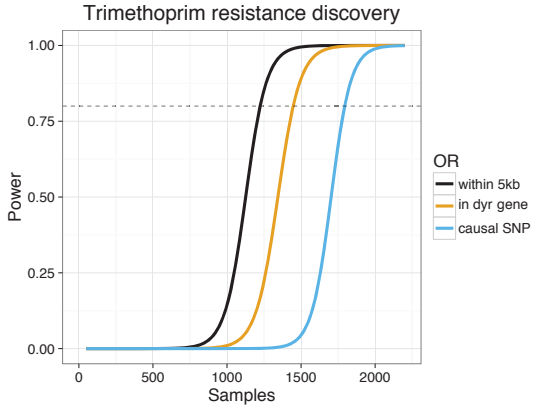
associated with the phenotype.



(a) Gene presence/absence at different odds-ratios.



(b) Using all informative k-mers versus a single length for gene mediated association.



(c) Detecting k-mers near, in the correct gene, or containing the causal variant for trimethoprim resistance.

Figure 2.6: Using simulations and subsamples of the population as described, power for detecting associations. All curves are logistic fits to the mean power over 100 subsamples.

Having knowledge of the true alignments, I then artificially associated an accessory gene with a phenotype over a range of odds-ratios and evaluated power at different sample sizes (fig. 2.6a). The expected pattern for this power calculation is seen, with higher odds-ratio effects being easier to detect. Currently detected associations in bacteria have had large effect sizes (OR $>$ 28 host-specificity (Sheppard et al., 2013); OR $>$ 3 beta-lactam resistance (Chewapreecha, Marttinen et al., 2014)), and the required sample sizes predicted are consistent with these discoveries.

The large k-mer diversity, along with the population stratification of gene loss, makes the simulated estimate of the sample size required to reach the stated power conservative. Convergent evolution along multiple branches of a phylogeny for a real population reacting to selection pressures will reduce the required sample size (Farhat et al., 2013).

I also compared the performance when using k-mers counted at constant lengths by DSK (Rizk et al., 2013) to perform the gene presence/absence association. Counting all informative k-mers rather than a pre-defined k-mer length gave greater power to detect

associations, with 80% power being reached at around 1 500 samples, compared with 2 000 samples required by 31-mers (fig. 2.6b). The slightly lower power at low sample numbers is due to a stricter Bonferroni adjustment being applied to the larger number of DSM k-mers over the DSK k-mers. This is exactly the expected advantage from including shorter k-mers to increase sensitivity, but as k-mers are correlated with each other due to evolving along the same phylogeny, using the same Bonferroni correction for multiple testing does not decrease specificity.

The strong LD caused by the clonal reproduction of bacterial populations means that non-causal k-mers may also appear to be associated. This is well documented in human genetics; non-causal variants tag the causal variant increasing discovery power, but make it more difficult to fine-map the true link between genotype and phenotype (Spain & Barrett, 2015). In simulations it is difficult to replicate the LD patterns observed in real populations, as recombination maps for specific bacterial lineages are not yet known. To evaluate the power of fine-mapping and associated locus to the single causal SNP I instead used the real sequence data and the effect size of a known causal variant, and evaluated the physical distance of significant k-mers from the variant site.

I tested the 68M k-mers from DSM for association with trimethoprim resistance: 2 639 k-mers reached significance, were mapped to a reference genome, and were found to cover most of the genome with a peak at the causal variant (fig. 2.7). I placed mapped k-mers near the correct physical location into three categories: those containing the causal variant I100L (10 k-mers), those within the same gene (74 k-mers), or those within 2.5kb in either direction (207 k-mers). Figure 2.6c shows the resulting power when random subsamples of the population are taken. As expected, power is higher when not specifying that the causal variant must be hit, as there are many more k-mers which are in LD with the SNP than directly overlapping it, thus increasing sensitivity.

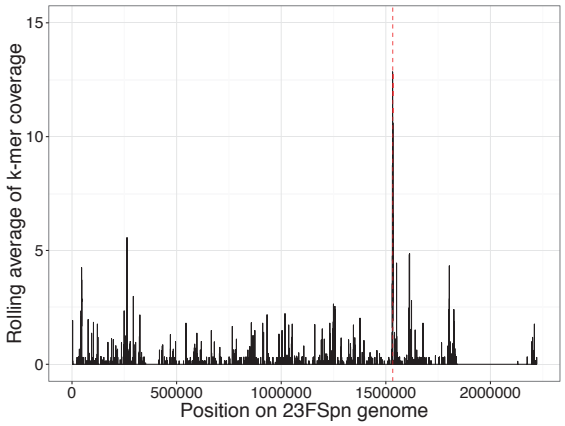


Figure 2.7: K-mers are mapped to the ATCC 700669 reference genome. Plotted coverage is the rolling average over 100bp windows over the genome. The red dashed line at 1 533 003bp shows the location of the causal variant, overlapping with the peak in coverage.

2.6.2 Antibiotic resistance in pneumococcal carriage

I then applied SEER to the sequenced genomes from the study described in section 2.6.1 (Chewapreecha, Harris et al., 2014), using measured resistance to five different antibiotics as the phenotypes: chloramphenicol, erythromycin, β−lactams, tetracycline and trimethoprim. Chloramphenicol resistance is conferred by the *cat* gene, and tetracycline resistance is conferred by the *tetM* gene, both carried on the ICE ICESp23FST81 in the *S. pneumoniae* ATCC 700669 chromosome (Croucher et al., 2009). For both of these drug resistance phenotypes the ICE contained 99% of the significant k-mers, and the causal genes rank highly within the clusters (table 2.2).

Antibiotic	Resistant samples	Number of significant k-mers			
		Total	Mapped to reference	Highest coverage annotation	Causal element
Chloramphenicol	204 (7%)	1 526	1 526	1 508 – ICE	166 – *cat*
				288 – ORF (UniParc B8ZK82)	
				206 – *rep*	
				166 – *cat*	
Erythromycin	803 (26%)	1 154	112	10 – permease (UniParc B8ZKV5)	4 – mega element
				8 – *prfC*	2 – *mef*
				6 – *gatA*	2 – omega element
				4 – ICE	
β−lactams	1 563 (51%)	23 876	17 453	381 – ICE	47 – *pbp2x*
				145 – prophage MM1	20 – *pbp2b*
				50 – SPN23F15110 (UniParc B8ZLE7)	8 – *pbp1a*
				49 – ICE *orf16*	
Tetracycline	1 958 (64%)	962	962	962 – ICE	96 – *tetM*
				136 – ICE *orf16*	
				121 – ICE *orf15*	
				96 – *tetM*	
Trimethoprim	2 553 (83%)	2 639	210	21 – *dyr*	21 – *dyr*

Table 2.2: Results from SEER for antibiotic resistance binary outcome on a population of 3 069 *S. pneumoniae* genomes. Significant k-mers were first interpreted by mapping to the ATCC 700669 reference genome. Up to the first four highest covered annotations are shown, and if the known mechanism is amongst these it is highlighted in orange. The ICE is the top hit in three analyses, as it carries multiple drug-resistance elements and is commonly found in multi-drug resistant strains (Croucher et al., 2009).

Resistance to erythromycin is also conferred by presence of a gene, but there are multiple genes that can be causal for this resistance: *ermB* causes resistance by methylating rRNA whereas *mef/mel* is an efflux pump system (Croucher, Harris, Fraser et al., 2011). In this population, this phenotype was strongly associated with two large lineages, making the task of disentangling association with a lineage versus a specific locus more difficult. I mapped some of the significant k-mers to the mega and omega cassettes, which carry the *mel/mef* and *ermB* resistance elements respectively.

I also mapped hits to other sites within the ICE, a permease directly upstream of *folP*, *prfC* and *gatA*. Macrolide resistance cassettes frequently insert into the ICE in *S. pneumoniae*, so it is in LD with the genes discussed above. In sulphamethoxazole resistance *folP* is modified by small insertions, with which the adjacent permease is in LD

with. Finally, *prfC* and *gatA* are both involved in translation, so could conceivably contain compensatory mutations when *ermB* mediated resistance is present. Further evidence of these compensatory mutations would be required to rule out the k-mers mapping to them simply being false positives driven by population structure.

Some k-mers did not map to the reference, as they are due to lineage specific associations with genetic elements not found in the reference strain. This highlighted both the need to map to a close reference or draft assembly to interpret hits described in section 2.4.2, as well as the importance of functional follow-up to validate potential hits from GWAS methods such as SEER.

Multiple mechanisms of resistance to $\beta-$lactams are possible (Chewapreecha, Marttinen et al., 2014). I considered just the most important (i.e. highest effect size) mutations, which are SNPs in the penicillin binding proteins *pbp2x*, *pbp2b* and *pbp1a*. In this case ranking annotations by highest coverage found these genes ranked top, but this was not sufficient evidence for discovery as so many k-mers were significant – either due to other mechanisms of resistance, physical linkage with causal variants or co-selection for resistance conferring mutations. Instead, I looked at the k-mers with the most significant p-values: the top four hit loci were *pbp2b* ($p = 10^{-132}$), *pbp2x* ($p = 10^{-96}$), putative RNA pseudouridylate synthase – UniParc B8ZPU5 ($p = 10^{-92}$) and *pbp1a* ($p = 10^{-89}$). The non-*pbp* hit is a homologue of a gene in linkage disequilibrium with *pbp2b*, which would suggest mismapping rather than causation of resistance.

Trimethoprim resistance in *S. pneumoniae* is conferred by the I100L mutation in the *folA*/*dyr* gene (Maskell et al., 2001). The *dpr* and *dyr* genes, which are adjacent in the genome, had the highest coverage of significant k-mers (fig. 2.8). To try and find the specific variant causal for the phenotype (i.e fine-mapping) I used the BLAT mapping of significant k-mers to a reference sequence, and called SNPs using bcftools (H. Li, 2011). I set quality scores for a read to be identical, as the Phred-scaled Holm-adjusted p-values from association. I then filtered for high quality ($QUAL > 100$) SNPs, and then annotated the predicted effect using SnpEff (Cingolani et al., 2012). I finally ranked the effect of missense SNPs on protein function using SIFT, which uses whether sites are conserved across the protein family to predict whether amino acid changes will alter protein function (Ng & Henikoff, 2003). Following this fine-mapping procedure, I called four high-confidence mutations that are predicted to be non-synonymous SNPs. One is the causal SNP, and the others appear to be hitchhikers in LD with I100L. The SIFT ranking places the known causal SNP top, showing that in this case fine-mapping is possible using the output from SEER.

I compared the performance of SEER to two existing methods. Chewapreecha, Marttinen et al. (2014) tested variants from a core-genome SNP mapping using plink (Purcell et al., 2007); population clusters were used to perform a CMH test to control for population structure. Sheppard et al. (2013) used fixed k-mer lengths of 21, 31 and 41 as counted

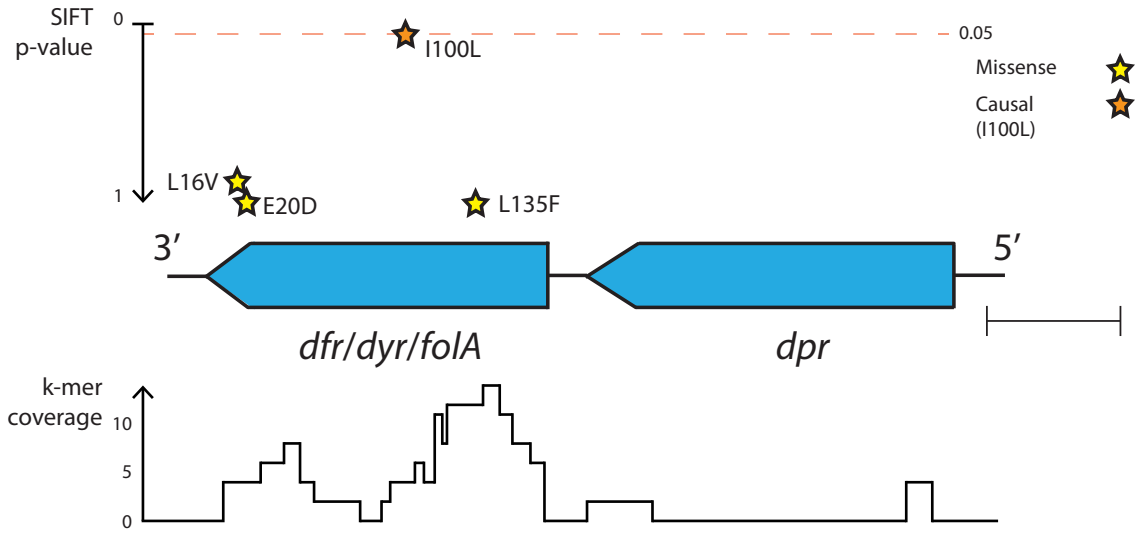


Figure 2.8: Fine mapping the causal variant for trimethoprim resistance. The locus pictured contains 72 significant k-mers, the most of any gene cluster (fig. 2.7). Coverage over the locus is pictured at the bottom of the figure. Shown above the genes are high quality missense SNPs, plotted using their p-value for affecting protein function as predicted by SIFT. Scale bar is 200 base pairs.

by DSK (Rizk et al., 2013), with a Monte Carlo phylogeny-based population control. As the second method is not scalable to this population size, I used the SEER population control as calculated from all genomes in the population and a subsample of 100 samples to calculate association statistics, which is roughly the number computationally accessible by this method. In both cases, the same Bonferroni correction is used as for SEER.

Antibiotic	Causal variant	Significant sites		Near correct site			Notes
		plink	dsk	plink			
Tetracycline	ICE, *tetM*	8 029	0	*tetM* – 124	ICE – 2240		
Chloramphenicol	ICE, *cat*	5 310	0	*cat* – 0	ICE – 1137		
β−lactams	*pbp2x*, *pbp1a*, *pbp2b*	858	0	*pbp2x* – 210	*pbp1a* – 113	*pbp2b* – 81	
Trimethoprim	*dyr* (I100L)	4 009	0	*dyr* – 47	*dpr* – 53		Causal SNP ranked 22nd
Erythromycin	*ermB*, *mef*, *mel*, *mefA*	8 469	0	None			Element not present in reference

Table 2.3: The power to find genetic associations with antibiotic resistance in the Maela study using existing methods. For each of the five antibiotics, the true causal variant is listed, as are the number of hits passing the significance threshold for each method (plink and DSK) and the number which map to the correct region.

Both SEER and association by core mapping of SNPs (using plink) identified resistances caused by presence of a gene, when it was present in the reference used for mapping (table 2.3). Both produced their most significant p-values in the causal element, though SEER appeared to have a lower false-positive rate. However, as demonstrated by chloramphenicol resistance, if not enough SNP calls are made in the causal gene this hinders fine-mapping. SNP-mediated resistance showed the same pattern since many other SNPs were ranked above the causal variant. In the case of β−lactam resistance both methods seem to perform equally well, likely due to the higher rate of recombination and the creation of mosaic *pbp* genes.

Additionally, as for erythromycin resistance, when an element is not present in the

reference it is not detectable in SNP-based association analysis. In such cases multiple mappings against other reference genomes would have to be made, which is a tedious and computationally costly procedure. Since the k-mer results from SEER are reference-free, the computational cost of mapping reads to different reference genomes is minimised as only the significant k-mers are mapped to all available references. Alternatively, the significant k-mers can be mapped to all draft assemblies in the study, at least one of which is guaranteed to contain the k-mer, to check if any annotations are overlapped.

The small sample, combined with fixed length 31-mer, approach did not lead to any words reaching significance for chloramphenicol, tetracycline or trimethoprim as the effect size of any k-mer is too small to be detected in the number of samples accessible by the method. I found 19 307 hits for erythromycin, and 419 hits for $\beta-$lactams, at between 1-2% MAF which are all false positives that would likely have been excluded by a fully robust population structure correction method such as the one the authors originally used.

2.6.3 Virulence of *Streptococcus pyogenes*

Most bacterial GWAS studies to date have searched for genotypic variants that contribute towards or completely explain antibiotic resistance phenotypes. As a proof of principle that SEER could be used for the discovery stage of sequence elements associated with other clinically important phenotypes, I applied the tool to 675 *S. pyogenes* (group A *Streptococcus*) genomes obtained from population diversity studies for genetic signatures of invasive propensity.

347 isolates of *S. pyogenes* collected from Fiji (Steer et al., 2009) were sequenced on the Illumina HiSeq platform, which I then combined with 328 existing sequences from Kilifi, Kenya (Seale et al., 2016). I defined those isolated from blood, CSF or bronchopulmonary aspirate as invasive (n = 185), and those isolated from throat, skin or urine as non-invasive (n = 490). I then ran SEER to determine k-mers significantly associated with invasion, followed by a BLAST of the k-mers with the nr/nt database to determine a suitable reference for mapping purposes.

After this preliminary analysis, I found the top hit was the *tetM* gene from a conjugative transposon (*Tn*916) carried by 23% of isolates (fig. 2.9a). These elements are known to be variably present in the chromosome of *S. pyogenes* (Roberts & Mullany, 2009), and the lack of co-segregation with population structure explained the power to discover the association. However, as a different proportion of the isolates from each collection were invasive (Fiji – 13%; Kilifi – 43%), the significant k-mers will also include elements specific to the Kilifi dataset. Indeed, I found that this version of *Tn*916 was never present in genomes collected from Fiji. To correct for this geographic bias, I repeated the SEER analysis by including country of origin as a covariate in the regression. This analysis removed *tetM* as being significantly associated with invasiveness, and highlighted the

importance of such covariate considerations in performing association studies on large bacterial populations.

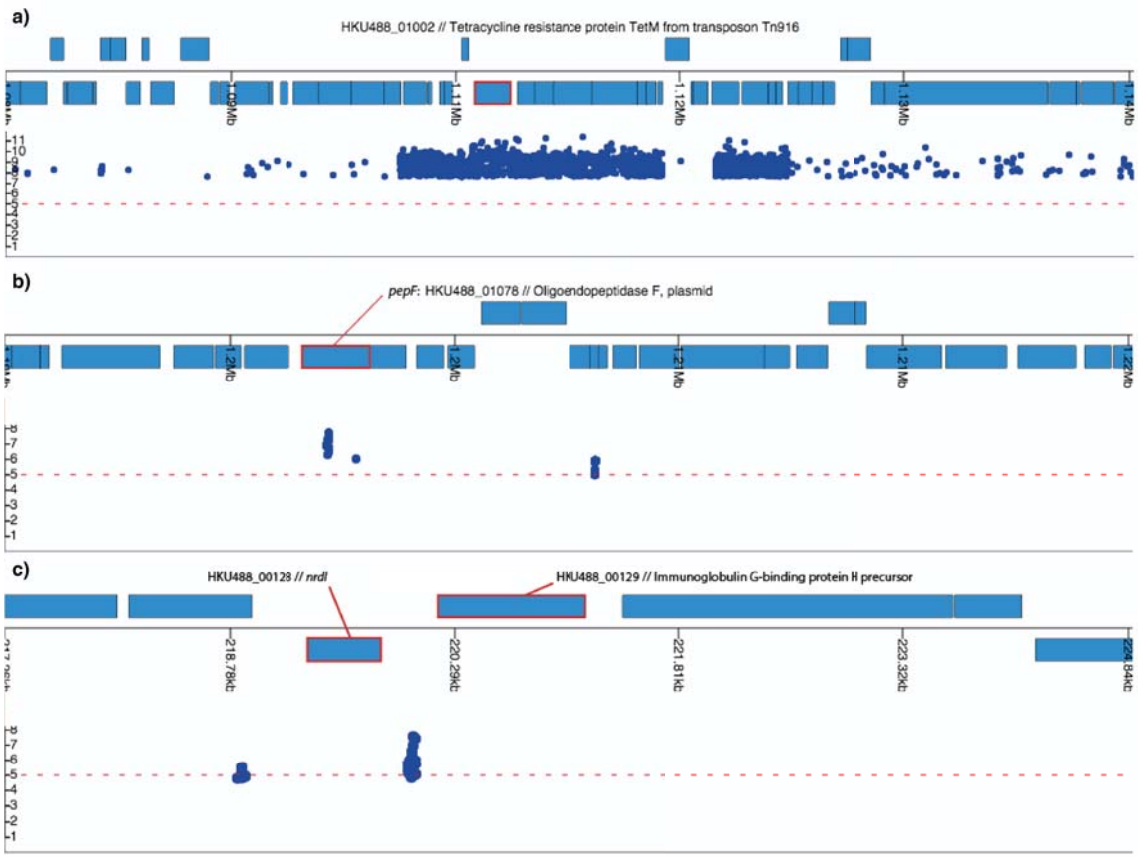


Figure 2.9: Phandango view of *S. pyogenes* HKU488 reference genome (blue blocks at top genes on forward and reverse strands, *tetM* highlighted in red) and Manhattan plot of start positions of significant k-mers: **a)** associated with invasiveness when not adjusted for country of origin; **b)** and **c)** adjusted for country of isolation.

After applying this correction, I identified two significant hits (fig. 2.9b,c). The first corresponded to SNPs associating a specific allele of *pepF* (Oligoendopeptidase F; UniProt P54124) with invasive isolates. This could indicate a recombination event, due to the high SNP density and discordance with vertical evolution with respect to the inferred phylogeny (Dubnau, 1999; Lefébure & Stanhope, 2007). The second hit represented SNPs in the intergenic region upstream of both IgG-binding protein H (*sph*) and *nrdI* (ribonucleotide reductase). In support of these findings, previous work in murine models have found differential expression of *sph* during invasive disease (Raeder & Boyle, 1993, 1995; T. C. Smith et al., 2003b), but little to no expression outside of this niche (T. C. Smith et al., 2003a). If these k-mers were found to affect expression of the IgG-binding protein, this would be a plausible genetic mechanism affecting pathogenesis and invasive propensity (Walker et al., 2014). The association of both of these variations would have to be validated either in vitro or a within a replication cohort, and functional follow-up such as RNA-seq may also help with determining the role of these genetic variants in

S. pyogenes pathogenesis.

In contrast, when I applied existing association methods described above (plink and DSK) to this *S. pyogenes* population dataset I found no sites significantly associated with invasiveness. The CMH test (stratified by BAPS cluster) that uses SNPs called against a reference sequence failed to identify the *tetM* gene and transposon as these elements are not found in the reference sequence. Furthermore, the population structure of this dataset is so diverse that 88 different BAPS clusters were found, which overcorrected for population structure when using the DSK method, leaving too few samples within each group to provide the power to discover associations.

2.7 Conclusions

SEER is a reference-independent, scalable pipeline capable of finding bacterial sequence elements associated with a range of phenotypes while controlling for clonal population structure. The sequence elements can be interpreted in terms of protein function using sequence databases, and I have shown that even single causal variants can be fine-mapped using the SEER output.

My use of all informative k-mers less than 100 bases long, a robust regression protocol and the ability to analyse very large sample sizes showed improved sensitivity over existing methods. This provides a generic approach capable of analysing the rapidly increasing number of bacterial whole genome sequences linked with a range of different phenotypes. The output can readily be used in a meta-analysis of sequence elements to facilitate the combination of new studies with published data, increasing both discovery power and confirming the significance of results.

As with all association methods, the approach is limited by the amount of recombination and convergent evolution that occurs in the observed population, since the discovery of causal sequence elements is principally constrained by the extent of LD. However, by introducing improved computational scalability and statistical sensitivity SEER improved on previous GWAS methods for answering important biologically and medically relevant questions.

In subsequent chapters I will start by using the GWAS techniques developed here to assess the contribution of bacterial variation to various stages of pneumococcal infection.