Pool-seq analysis for the identification of polymorphisms in bacterial strains and utilization of the variants for protein database creation

Rigbe G. Weldatsadik

School of Science

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 10.10.2016

Thesis supervisor:

Prof. Harri Lähdesmäki

Thesis advisor:

Adj Prof. Sakari T. Jokiranta

Aalto University
School of Science

AALTO UNIVERSITY
SCHOOL OF SCIENCE

ABSTRACT OF THE
MASTER'S THESIS

Author: Rigbe G. Weldatsadik

Title: **Pool-seq analysis for the identification of polymorphisms in bacterial strains and utilization of the variants for protein database creation**

Date: 10.10.2016 Language: English Number of pages: 8+60

Department of Computer Science Professorship: T-61

Supervisor: Prof. Harri Lähdesmäki Advisor: Adj Prof. Sakari T. Jokiranta

Pooled sequencing (Pool-seq) is the sequencing of a single library that contains DNA pooled from different samples. It is a cost-effective alternative to individual whole genome sequencing. In this study, we utilized Pool-seq to sequence 100 *streptococcus pyogenes* strains in two pools to identify polymorphisms and create variant protein databases for shotgun proteomics analysis. We investigated the efficacy of the pooling strategy and the four tools used for variant calling by using individual sequence data of six of the strains in the pools as well as 3407 publicly available strains from the European Nucleotide Archive. Besides the raw sequence data from the public repository, we also extracted polymorphisms from 19 *S.pyogenes* publicly available complete genomes and compared the variations against our pools.

In total 78955 variants (76981 SNPs and 1725 INDELs) were identified from the two pools. Of these, $\sim$ 60.5% and 95.7% were discovered in the complete genomes and the European Nucleotide Archive data respectively. Collectively, the four variant calling tools were able to mine majority of the variants, $\sim$ 96.5%, found from the six individual strains, suggesting Pool-seq is a robust approach for variation discovery. Variants from the pools that fell in coding regions and had non synonymous effects constituted 24% and were used to create variant protein databases for shotgun proteomics analysis. These variant databases improved protein identification in mass spectrometry analysis.

Keywords: Pool-seq, *streptococcus pyogenes*, variant protein database

Preface

I would like to thank Harri Lähdesmaki for supervising this thesis work and the valuable feedback he has given me. I would also like to extend my gratitude to my advisor and research group leader Sakari Jokiranta for giving me the chance to work on this project and for all the insightful and fruitful discussions we have had. I am also thankful to all the members in Jokiranta's group as well as Juha Kere, Jaana Vuopio and Markku Varjosalo and all their team members.

My special thanks goes to my husband Dawit and my family (especially my mom, Tsega Kahsay and my dad, Gebremichael Weldatsadik) for the unwavering support and love they have always given me, especially during those trying times of my life. Thank you mom and dad for instilling in me values I hold dear and for being my constant source of inspiration. Thank you Dawit for dedicating your research work to my cause and for being a wonderful partner in this journey.

And finally, I would like to thank Päivi Koivunen, the program study coordinator, for patiently handling all my inquiries related to the thesis writing process and also other study matters. The extensive resources I required for completing this thesis work was provided by the IT center for science(CSC). Many thanks to the CSC team for this valuable infrastructure and all their assistance.

Otaniemi, 10.10.2016

Rigbe G. Weldatsadik

Contents

Symbols and abbreviations

Abbreviations

Pool-seq	Pooled Sequencing
ENA	European Nucleotide Archive
GAS	Group A Streptococcus
NGS	Next Generation Sequencing
GWAS	Genome Wide Association Studies
SNP	Single Nucleotide Polymorphism
INDEL	Insertions and Deletions
MNP	Multi-Nucleotide Polymorphism
MS	Mass Spectrometry
RAD	Restriction-site Associated DNA markers
OMIM	Online Mendelian Inheritance in Man
PCR	polymerase chain reaction
PMD	Protein Mutant Database
PHI	Pathogen - Host Interaction Database
EST	Expressed sequence TAG

Tables and Figures

List of Tables

List of Figures

1. Introduction

Bacteria are ubiquitous, found in diverse niches, including humans and other organisms, and extreme environmental conditions. Humans have 10 times more bacteria than cells in their bodies. Most of these bacteria are harmless and some are even beneficial, while others are pathogenic. The first bacteria to be whole genome sequenced was Haemophilus Influenza in 1995 [29]. Since then, around 20,000 bacterial genomes of 50 different phyla have been sequenced and made available to the public. This explosive increase has been possible due to the vast price reduction of the Next-generation sequencing (NGS) technologies.

We have come to understand through such sequencing efforts that bacteria are very diverse and for certain bacteria, even within the same species, there could be a lot of variation in terms of gene content and genome size. This suggests sequencing of large number of samples, such as different strains of a bacteria, is vital for various population based studies. Even though NGS technologies offer a substantial cost reduction compared to previous methods such as Sanger sequencing, the cost associated with whole genome sequencing of large cohorts is still prohibitive for many labs. As a result, various cost effective alternatives such as Pool-seq, RNA-seq, RAD-seq and exome sequencing have been utilized in various studies.

Pool-seq is the sequencing of a single library containing equal amounts of DNA from different samples. The samples could be tagged (barcoded) before pooling which will enable to distinguish the sequence reads coming from the different individuals. However, tagging incurs additional effort and cost when attaching and demultiplexing the barcodes during sample preparation and analysis, especially for large number of samples. For this reason, samples are usually pooled without tagging and this strategy is also adopted in the current study. Pool-seq has been effectively utilized in studies that involve allele frequency estimation and polymorphism identification of large sample of organisms. In the current study, Pool-seq is used to study genetic polymorphisms in 100 strains of one of the most important human pathogens, *Streptococcus pyogenes* bacteria. *S.pyogenes* or Group A streptococcus (GAS), commonly known as the flesh eating bacteria, is a Lancefield group A bacterial pathogen that causes a multitude of non-invasive and invasive infections and post-infection sequelae throughout the world. Diseases caused by GAS include pharyngitis,

impetigo, erysipelas, cellulitis, necrotizing fasciitis, scarlet fever, toxic shock syndrome, rheumatic fever and glomerulonephritis. GAS is part of the normal flora in humans typically in the respiratory tract, and can be considered an opportunistic pathogen.

Besides the Lancefield antigens, the M protein surface antigen is also used for classifying GAS. Currently more than 100 distinct M serotypes have been identified. Certain M types such as M1 and M3 are mostly isolated from both human invasive infections and pharyngitis in high-income countries [70]. In this study, 50 non-invasive and 50 invasive strains were sequenced in two pools. The publicly available M1 strain, SF370, was used as a reference genome to explore polymorphisms of these strains in the two pools.

In addition to the discovery and exploration of polymorphisms from pooled sequence data, creation of variant protein databases for shotgun proteomics analysis of GAS strains is central to this study. Protein databases are used in shotgun proteomics studies for searching the mass spectra identified from experiments against the theoretical mass of the peptides found in the databases. Such databases usually contain wild type proteins and therefore fail to identify variant peptides. Various studies have devised ways to enable the consideration of variation information in database searches. For instance, certain search engines have an option to search for all possible amino acid substitutions due to SNVs. In addition, several authors have used combinatorial approaches ('shotgun annotation') as well as known variants in humans (including cancer-specific ones) to create such variation databases.

To our knowledge, this study is the first to utilize Pool-seq for polymorphism identification as well as create a variant proteome database for GAS. The genetic polymorphisms in the pooled samples were mined using 4 variant calling tools, whose efficiency was evaluated based on results from individual sequence analysis of 6 strains that were in the pools and also publicly available data. A custom in-house script was utilized to create variant protein databases from the two pools.

GAS and other pathogen microbes with high species-level genetic diversity would benefit from large sample studies and findings from this study, which includes examining the efficiency of Pool-seq , will motivate the use of the method for addressing important study questions in various non model organisms. In addition, the availability of variant protein databases for this essential pathogen, we anticipate,

will enable the identification of novel peptides in shotgun proteomics studies which may have implications in clinical applications.

This thesis work is part of a project that aims to identify novel candidate antigens for clinical diagnosis of GAS, GGS, GBS and GCS pathogens. The author is solely responsible for the sequence data analysis and database creation, with out undertaking any of the laboratory work in this project. The rest of this thesis is organized as follows. In Chapter 2, I describe briefly streptococcal diseases, NGS technologies and downstream analyses, issues that concern Pool-seq and Pool-seq variant analysis and bottom up shotgun proteomics analysis and the methods employed for mass spectral search. The materials and methods employed in this study and the results from the variant detection analysis as well as the database creation process are presented in chapters 3 and 4 respectively. In the final chapter, chapter 5, a discussion of the various results obtained and the conclusions drawn in the course of this study are reported.

2. Background

This chapter gives an overview of the various issues that are addressed in this thesis work. The first part briefly introduces how streptococci bacteria are classified and the diseases they inflict on humans. This is followed by a general description of NGS technologies and downstream analysis steps. The next two parts discuss important considerations in relation to using Pool-seq for polymorphism identification and allele frequency estimation, and introduce variant calling tools suitable for Pool-seq studies respectively. Then follows a few examples of Pool-seq applications. The last part describes the bottom-up shotgun proteomics set up with an emphasis on variant protein databases.

2.1 Streptococcal diseases

The *streptococcus* genus is comprised of spherical (coccus) bacteria that stain purple in the Gram stain test, also known as Gram-positive bacteria. Currently, there are about 115 species under the streptococcus genus [41]. These species are classified into alpha, beta and gamma based on their hemolytic properties, i.e. the area of the blood agar around the colony becomes dark and greenish, lightened yellow (transparent) and unchanged respectively. The beta hemolytic streptococci are further serotypically classified according to the Lancefield carbohydrate group present on the bacterial cell wall. The Lancefield groups A, B, C, D, and G (GAS, GBS, GCS, GDS, GGS) have been known to cause infections in humans.

GAS causes the most devastating human infections through out the world, which vary in clinical spectra and severity, causing at least 517,000 deaths annually [13]. The prevalence of severe GAS diseases is estimated to be 18.1 million, with 1.78 million new cases per year. These severe cases include rheumatic fever, rheumatic heart disease, post-streptococcal glomerulonephritis, and invasive infections such as bacteraemia. The incidence of less severe diseases such as strep throat (pharyngitis) is estimated to be 616 million cases per year. GBS mainly causes pneumonia and meningitis in newborns and the elderly; GGS on the other hand has similar spectrum as that of GAS. In Finland, an increase in GGS bacteraemia cases has been observed recently [61].

Certain features of a bacteria enables it to attach to, colonize and invade the host cell and are associated with virulence. For instance in GAS, the hyaluronic capsule mimics the mammalian polysaccharide which enables it to avoid detection by the host immune system. The surface M protein is another virulence factor in GAS that has antiphagocytic properties. The N terminal part of this protein is hypervariable and is the basis for the serological classification of GAS strains. Based on this typing, more than 100 distinct strains have been identified so far.

Diagnosis of streptococcal diseases is mainly based on culture. Antigen detection methods are also being used in certain countries. The antigen based diagnostic methods are quicker and also relatively reliable and thus can be used as point of care tests. However, there is still a need for more sensitive and specific tests; for instance, there are no current tests that recognize GGS antigens resulting in failure to diagnose substantial numbers of streptococcal tonsillitis cases.

2.2 Next generation sequencing and downstream analyses

DNA sequencing of various organisms has been crucial to numerous kinds of studies including sequence variation detection and interpretation. Compared to the previous methods such as the capillary electrophoresis based Sanger sequencing which produces 96 sequencing reads, NGS (also referred to as high-throughput sequencing, i.e.HTS) technologies produce large number of sequence reads per experiment [39] which are shorter in length and lower in quality. NGS technologies are cheaper, quicker and need significantly less DNA. There are a number of NGS technologies including Illumina (Solexa), Roche 454, Ion torrent and SOLiD. These technologies differ in the protocols they employ during template preparation, sequencing and imaging, and data analysis. But all of these methods involve random shearing of the DNA in to smaller sized templates and the immobilization of the templates to a solid surface. The templates' nucleotide sequences are inferred from the light signals emitted (recorded using a camera) during the incorporation of the bases via synthesis or ligation. This process is carried out in parallel resulting in the production of thousands to millions of short reads. All of the NGS platforms introduce sequencing errors that pertain to the unique combination of protocols they each follow. The sequence reads, in downstream analysis applications, will either be used to reconstruct

the underlying whole genome de novo or will be aligned to a reference genome, for example to identify polymorphisms.

Various alignment tools exist that are capable of mapping the large number of reads in a reasonable amount of time. To speed up the alignment process, they use either a hash table (e.g Novoalign[1], stampy[52], SHRiMP[65]) or prefix/suffix array trees (e.g. BWA[48], Bowtie[44]) to index the reference genome or the read sequences. In the hash table implementation, k long substrings (k-mers) of the query are created and saved in a hash look up table against which seeds are searched. The candidate seeds will then be extended using a local alignment algorithm such as the Smith-Waterman algorithm[68]. In the second category, one of the most common implementations is the FM-index. This is based on the Burrows-Wheeler transformation (BWT) [12] where the rotated instances of the string are lexicographically sorted and the last columns represent the transformation. Then using the FM-index a last-to-first column mapping can be done on this transformation. The BWT-based aligners are faster and more memory efficient but less sensitive than the hash-based methods.

After the alignment step, positions that show evidence of variation are determined in what is called the variant calling step. The variants can be small, usually less than 50bp, as in SNVs and short INDELs or large ones, such as structural variants (SVs). Due to the length of the reads produced from most NGS platforms, inferring larger variants could be challenging. Having a pair of reads using paired-end or mate-pair sequencing (in the former you sequence the two ends of a size-selected fragment while in the latter the sequence is first circularized with the ends tagged and then size-selected fragments that contain the ends are sequenced) with known distance and orientation between the pairs helps to get around the problem of short length reads. The polymorphic positions and frequencies of the variant alleles are central in many population genetics studies.

2.3 Pool-seq

The amount and quality of data that is being generated by NGS technologies is improving while at the same time the cost of sequencing is decreasing significantly. NGS technologies have basically democratized sequencing by making it affordable to a large number of investigators. As a result of this, whole genome sequencing of

various non model organisms has been possible; some organisms have had more than one type or strain sequenced [26, 34]. A decade ago, sequencing of a finished bacterial genome using Sanger sequencing could cost up to $50,000; currently, a draft sequence can be generated at a fraction of that [43]. In bacteria this has opened doors for vital population based studies such as investigating polymorphisms associated with antibiotic resistance mechanisms and evolutionary pathway of virulent clones (see [15, 55] for example). For GAS, currently 45 complete GAS sequences have been recorded in the Genomes OnLine Database [62].

Nevertheless, sequencing is still costly for many research groups wishing to undergo large scale studies that involve large number of individuals, for instance genome wide association studies (GWAS). For this reason, various strategies have been adopted that aim to reduce the sequencing cost while still being able to draw statistically meaningful conclusions from the data. Some of these strategies include:

- Exome sequencing: only the gene coding regions of the genome are captured and sequenced. This method has been widely used especially in disease association studies to identify causative variants found in coding regions. However, many studies have discovered associations between diseases and variants that fall in non coding regions, such as promoters. In addition, it is usually not possible to capture all the exons and building exon capture kits for different species is expensive.

- RNA-seq : involves the sequencing of mRNA transcripts, which are distinguished by their poly A tails. Here again the focus is on the gene coding regions but unlike exome sequencing, only expressed transcripts are sequenced and no special capture kits are needed.

- RAD-seq : only regions that flank a restriction site are sequenced. This method relies on linkage disequilibrium (LD) and may give a biased allele frequency estimate if the restriction site contains polymorphisms that are in LD with nearby snps.

Another approach that has been employed to mitigate the high cost associated with large sample studies is Pool-seq. While all of the above strategies attain cost reductions by sequencing only part of the genome, in Pool-seq the whole genomes

of multiple samples is sequenced. Pool-seq achieves cost effectiveness through the sampling of multiple individuals and therefore limiting the amount of redundant reads to be considered. Pool-seq can provide better allele frequency estimates than individual sequencing if large number of samples are included in a pool. This is because sampling variance decreases as the sample size increases and allele frequencies are usually estimated from samples drawn from a larger population. It is possible to tag (index) sequences to identify the individuals in a pool but usually this method incurs more cost and therefore unindexed pooled sequencing is mostly used.

In Pool-seq, there is an assumption that the pool is composed of equal DNA amounts from all the individuals. Uneven representation of individuals in a pool could lead to significant allele frequency differences and therefore large pool sizes are recommended to reduce the effect of individual read depth variation. Moreover, when the assumption of equal DNA contributions is violated, which happens more often than not, and sequencing error rates are high, Pool-seq fails to identify rare variants [20]. In addition, haplotype information is lost during pooling and Pool-seq is unsuitable for studies that rely on linkage disequilibrium.

Several studies have developed statistical theories for the analysis of pooled samples and to infer population genetics estimators such as Tagima's π and Watterson's θ from such data [27, 30, 32]. These studies emphasize the importance of sample size and read depth for efficient SNV identification and allele frequency estimation. Pool-seq could be more effective for such tasks than individual sequencing given large enough sample sizes and high read depth, for instance when the pool contains more than twice the number of individual sequences and the coverage per individual sequences is at least two [30].

2.3.1 Examples of Pool-seq applications

Pool-seq has been used in studies of various nature that involve the sequencing of large number of samples such as GWAS and/or for which it is difficult to obtain individual samples, as in cancer and metagenomics. The following are but a few examples of such studies. For more examples and a comprehensive review of Pool-seq and guidelines to follow see [67] and the references thereof.

- GWAS. The common variant common disease paradigm has been successful in identifying causal variants with small effect sizes for various diseases. However, since the common variants explain only a small proportion of heritability, there is now a shift to investigate rare variants as well. Large sample sizes are required to detect associations in such cases and individual sequencing is still expensive, and therefore various studies have employed Pool-seq instead. Using Pool-seq, rare novel pathogenic mutations were found in the PSEN1, GRN and MAPT genes in Alzheimer's disease [40]. Similarly, Pool-seq was used to identify three rare variants in Crohn's disease associated genes [38].

- Evolve and resequence (E&R). These experimental evolution studies, especially those that start from a segregating population, typically involve Pool-seq as genome wide polymorphism data is required. Similar to GWAS, E&R studies aim at genotype-phenotype mapping. Unlike GWAS however, the studies are carried out under researcher controlled environments and conditions. Most E&R studies have been carried out on D. melanogaster. For instance, in a longevity study in flies, 156 genes were found to be divergent between flies that have been selected for 50 generations and unselected flies [63].

- Reverse ecology. Pool-seq is valuable in studies that use genomics to study ecological factors deriving selection. It has been used to study adaptive genetic variations in the herb Arabidopsis halleri from the Alps that are associated to climatic variations. 175 genes were found to be associated with the 5 tested climatic factors [28].

2.4 Variant calling of pooled data

One of the main differences between variant calling in pooled samples and that of individual samples is the frequency of the variant alleles in each individual. For instance, for diploid individuals, the variant allele frequency can only be 0 (reference homozygous), 0.5 (reference heterozygous) and 1 (variant homozygous), while for pooled samples the frequency could be any number between 0 and 1, with the minor allele frequency being $\frac{1}{h}$ for h haploid genomes. In individual sequencing data, at a depth 30x the variant allele will often be observed sufficiently that it is possible to make a reliable distinction between true variants and errors [8]. In Pool-seq on

the other hand, the minor allele could be found at a frequency equal to or below the sequencing error rate. In light of this fact, pooled variant calling tools employ different combinations of methods including error models, read quality and various characteristics of the sequencing errors (such as lower base quality, strand bias and clustering at certain positions in the read [7]) to make a reliable detection of rare variants. For instance, while tools that are not suited for pooled samples fail to call a variant that has a frequency of less than $\sim 0.5 \times depth$ at a certain position, pooled variant calling tools would more or less be able to identify a rare variant that is represented in $\sim \frac{1}{h} \times depth$ of the reads. This suggests using tools that are not geared towards the analysis of pooled data may result in many true variants being confounded with sequencing errors and therefore not being called.

There are a number of tools available for variant detection of pooled data. Some of them use Bayesian approaches with pre-specified prior probability of observing a variant (GATK, MAQ) and there are those that use heuristics methods, such as the minimum allele frequency threshold, for filtering candidates (VarScan) and others that use frequentist approaches (SNVer). There are those (such as snape, SNPSeeker) that can only identify SNPs while majority are capable of identifying SNPs and INDELs. While most of them use a constant error rate depending on the sequencing platform, tools such as EM-SNP and LoFreq model the error rate in a position specific manner.

The four variant calling tools used in this study employ both bayesian and frequentist approaches. SAMtools [49] is designed for diploid indviduals and uses a Bayesian model with a binomial likelihood and a pre-specified prior probability (for a heterozygote, 0.001 for the discovery of new SNPs and 0.2 at known SNP sites) to determine the posterior probabilities of the three possible genotypes. GATK's Unifiedgenotyper [23] uses the same model as SAMtools but partitions the posterior probability over all the possible genotypes and also uses various filtering strategies. Freebayes [31] also employes a Bayesian model but operates on haplotypes from a local de-novo assembly of reads than single positions, unlike SAMtools and Unifiedgenotyper. On the contrary, SNVer [74] uses a frequentist approach (variant calling as a hypothesis testing problem) to determine for each pool the p-value cutoff of true variants based on a binomial model of variant allele frequency and sequencing error. It then combines the p-values from individual pools to give an overall p-value; unlike in the other

methods where the decision is dichotomous (is or is not a variant), ranking of the variants is possible. Excepting SAMtools, all three tools have options for the analysis of pooled samples. We included SAMtools for comparative purposes as it is a popular tool and we have come across a Pool-seq study that employed it.

There are other variant calling tools for Pool-seq data but we chose these 3 based on popularity, convenience and their previously reported performance. Below is a list of some of the other tools that enable variant calling from pooled samples.

- CRISP [7]: Uses contingency tables to compare the distribution of allele counts across multiple pools in order to identify rare variants. It can call SNVs and short INDELs. It is not capable of calling variants from a single pool but requires multiple pools. It accepts SAM/BAM files as inputs and outputs a VCF. It can be used with Illumina and Solid reads. It is reported to be better than Varscan and SNPSeeker.

- Syzygy [64]: Uses a likelihood computation to determine if a position contains a non-reference allele given all the alleles in a pool using Baye's rule. It can call SNVs and short INDELs. Accepts additional pool/target info files besides SAM/BAM input but the output is a CSV format than VCF. It can handle Illumina and Solid reads.

- SPLINTER [72] : It is an extension of the variant caller SNPSeeker [24](which calls only SNPs). It compares observed allele frequency and distribution of sequencing errors using Kullback-Leibler (KL) distance. It needs an additional positive control (besides the negative control data, such as a plasmid DNA, required in SNPSeeker). It can call SNVs and INDELs. Accepts only SCARF input and outputs CSV. It can not handle Solid reads. There is no dedicated download page available, it requires registration.

- Varscan [42]: Applies heuristic filters (for instance, minimum number of supporting reads and allele frequency threshold) to each candidate site. It can identify SNVS and INDELS. Accepts pileup input rather than SAM/BAM. Outputs both VCF and CSV files. It can also call variants from exome and RNA-seq data. Accepts Illumina, Solid and Roche/454 reads.

- LoFreq [75]: Uses Poisson-binomial to model the distribution of variants and the Phred base quality scores to model the errors. It can call SNVs and INDELS.

It accepts SAM/BAM inputs and outputs VCF. It does not call genotypes and also does not process inputs from different pools simultaneously. It can call variants from exome and targeted resequencing data.

- vipR [4]: Uses the Skellam distribution to identify sites with significant difference in minor allele frequencies in at least two pools. It calls SNVs and INDELs. Accepts only pileup input and outputs VCF. It requires multiple pools. It can handle Illumina and Solid reads.

2.5 Variant protein database creation

Proteomics is the study of proteins in a holistic manner including their biological functions, processes and interactions. One of the main goals in proteomics is identification and quantification of a species' proteome. Bottom-up mass spectrometry (MS)-based shotgun proteomics has been widely utilized for this purpose, especially in large scale studies.

Bottom-up shotgun proteomics involves the digestion of a mixture of proteins into peptides by proteolytic enzymes (such as trypsin), followed by MS analysis that generates mass spectra. The experimental mass spectrum is then analyzed to identify the peptides. Figure 1 depicts the steps involved in a bottom-up shotgun proteomics including the creation of the experimental mass spectrum and the identification of peptides from the spectrum.

The reliable identification of peptides/proteins depends on various issues ranging from the sample preparation and MS instrument (and fragmentation techniques) to the computational methods employed. Identification of the peptide sequences can be accomplished in 4 ways, by matching the spectra to a protein sequence database or a spectra database, by using sequence tags or through de novo sequencing (see part 2 of Figure 1).

Utilization of sequence databases is the most popular method in which the acquired experimental spectrum is compared to the theoretical spectrum obtained from the *in silico* digested sequences found in the database. This comparison is performed by search engines that score peptides, i.e. based on how similar they are to the experimental spectrum, using various criteria such as the parent ion mass tolerance,

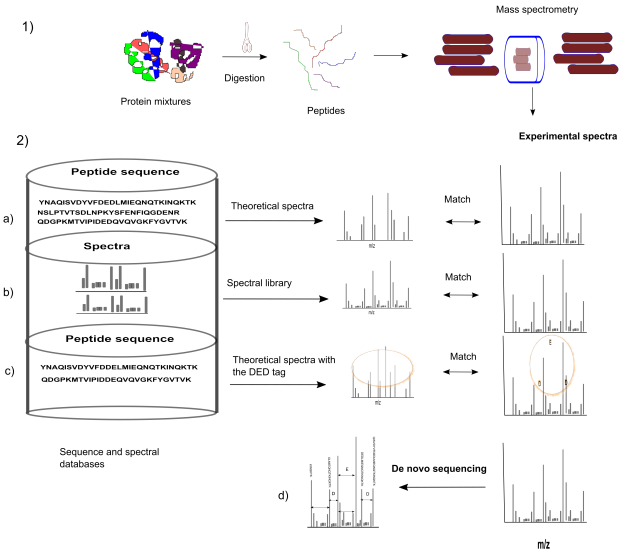


Figure 1: Bottom-up shotgun proteomics steps. 1)Proteins will be digested in to peptides and subjected to tandem mass spectrometry analysis. 2) The experimental spectra acquired will then be used to identify peptides using four methods. a)Using a sequence database in which theoretical spectra obtained via *in silico* digestion of the sequences are matched to the experimental spectra. b) Using a spectral database where the spectra are matched to the experimental spectra. c) Using tag short sequences and a sequence database in which only theoretical spectra of peptides that contain the tag sequence are matched to the experimental spectra and d) The peptides are identified de novo from the experimental spectra without using a database.

digestion enzymes, post-translational or chemical modifications and fragment ions expected. The search tools vary and the concordance among them when applied to the same dataset is between 70-80%. There are various protein sequence databases available for different organisms including Entrez Protein sequence database, Refseq, Uniprot and the International Protein Index(IPI) and they vary in terms of quantity and quality. For instance, Refseq and Uniprot are better annotated while Entrez contains larger number of sequences.

Not all the peptide to spectrum matches (PSMs) reported by search tools are true due to a number of reasons such as the low mass accuracy of certam MS instruments. The confidence of the PSM is assessed by the search score which is converted and reported as a p-value or an E-value in some tools. Multiple test correction is required when applying these single spectrum scores to multiple spectra analysis. The most common method used for this purpose is the false discovery rate (FDR). The FDR is usually estimated by using a decoy database which contains the reversed or shuffled protein sequences.

Instead of sequence databases, previously deposited spectra, against which the acquired spectra are searched, can also be utilized. This approach has an advantage over sequence databases in terms of speed, error rates and sensitivity. However it is limited by the availability of previously analyzed spectra.

On the other hand, de novo sequencing involves the determination of sequences directly from the spectra. As it does not require sequence or spectral databases, it can identify peptides that are not present in the databases. Yet, it is computationally intensive and is therefore mostly used when a protein sequence database is unavailable or to interrogate spectra unassigned during the database search. It can also be used to validate results obtained using databases.

Tag sequences are hybrid methods in which a short sequence tag is inferred de novo followed by a database lookup of the tag sequence together with the sequence masses flanking it. The database search in this case will be restricted only to those peptides that contain the sequence tags which will speed up the search. This approach is usually employed to identify post-translational and chemical modifications.

Not all the spectra from MS analysis can be matched to a sequence in a protein database owing to various reasons including post translational and chemical modifications, protein isoforms and amino acid substitutions (due to polymorphisms) [11, 19]. *Nesvizhskii et al.* [57] demonstrated that there are high quality spectra that remain unmatched after the first initial search against a conventional database and that using several types of databases leads to increased identification of peptides, including those that contain modifications and polymorphisms. Moreover, *Dasari et al.* reported that 3.2% to 7.1% of spectra showed evidence of mutations in the different samples they analyzed [21].

To address the issue of amino acid substitutions, certain database search engines such as Mascot and X! Tandem provide options to search for all possible amino acid substitutions that result from SNPs [19, 18]. However, such exhaustive search methods lead to an explosion of the search space and loss of sensitivity, at the same time requiring increased processing time [56]. It has also been shown that tag sequencing can be used to identify such mutations [21]. But, this method also suffers from computational inefficiencies especially for large scale studies.

Multi-stage strategies using the same tool iteratively or a combination of various

tools have also been used to decrease the number of unassigned spectra. For instance, in the first run, few or no modifications could be allowed in addition to proteolytic enzyme constraint. In the following runs, these criteria could be relaxed and a subset of the database could be searched. Alternatively, the searching could start by using multiple database search engines on sequence databases, followed by the utilization of spectral libraries or exhaustive searching.

A more commonly used alternative is the use of customized databases that incorporate known variants. For instance for humans, by utilizing different sources such as dbsnp, OMIM, PMD, and PHI, the SysPIMP and CanProvar databases have been developed [77, 50]. SysPIMP contains human disease related sequences which can be searched using MS data utilizing the X!tandem search engine. CanProvar contains cancer related variations from various sources as well as known coding variants from dbsnp. In this study, a custom database that contains variants identified from our next generation sequence analysis has been developed for GAS.

3. Materials and methods

This chapter contains an outline of the materials and tools I used in this study. The first part concerns Pool-seq variant calling, followed by a part that describes in detail the protein database creation process. For variant calling, I mostly used free existing software tools while for the protein database creation I developed a custom python script.

3.1 Bacterial strains

The 100 GAS strains used in the two pools are listed in the supplementary material (Table S1). The strains were selected from the bacterial culture collection of the National Institute of Health and Welfare so that each of the pools contained similar wide array of emm types isolated in wide geographical area in Finland within years 1995-2012.

3.2 Pooling and sequencing

All strains were cultured overnight at +35 °C on blood agar plates in 5% CO2. DNA was isolated using UltraClean Microbial DNA Isolation Kit (MoBio) according to manufacturer's instructions except for the following modifications: in the beginning 300 µl MicroBead solution and 6 µl mutanolysin (1 mg/ml) was mixed in a tube followed by addition of bacteria scraped with a 10 µl loop from the culture plate. After incubation for 60 min at +37 °C, the solution was transferred to a MicroBead tube, and 2 µl of RNAse A (1mg/ml) was added. From there, the manufacturer's instructions were followed until at step 18, 35 µl of solution MD5 was added followed by 2 min incubation. Before pooling, the quality and the integrity of the DNA was checked using Nanodrop equipment (Thermo Scientific) and agarose gel electrophoresis. From each GAS strain, 400 ng of DNA was used in one of the pools of 50 strains. The pools were precipitated and vaporized with SpeedVac and concentrations were measured with Qubit 62.5 ng/µl and 86.4 ng/µl for pool 1 and 2, respectively. Sequencing was done at Science for Life laboratory in Stockholm using Illumina HiSeq 2500 with approximately 20000x mean coverage. Each of the pools were sequenced in two lanes

using the paired-end sequencing strategy.

3.3 Public data

GAS sequences of 3407 runs and 20 complete genomes were downloaded from ENA and NCBI respectively. The runs from ENA were all paired-end sequenced using Illumina. The raw sequences of the reference strain SF370 are also available publicly but they were not included in the public data analysis.

3.4 Variant calling

The major steps involved in variant calling are summarized in Figure 2 and briefly described here. A number of tools exist that are capable of manipulating the data at each step of the variant calling workflow. The step that proceeds the alignment of the reads to the reference genome is a pre-processing step which involves checking the quality of the reads (using such tools as FastQC, SolexaQA) and trimming of possible adapter contaminants and low quality bases (using Trimmomatic, Cutadapt etc). FastQC [66] presents quality metrics such as per base sequence quality, per base GC content, over-represented sequences etc. in a user friendly graphical interface. FastQC (version 0.11.2) was used for quality inspection of the sequence reads from the pools and the ENA data. Trimmomatic [9] handles paired-end reads and can trim adapters and also low quality bases from the start and end of the read or using a sliding window. A custom python script was used to automatically extract adapter and primer contaminants from the FastQC output and trim them using Trimmomatic (version 0.33).

The alignment step is one of the most fundamental steps as the accuracy of the alignment is one factor that will influence the quality of the variants mined. The large number of short length reads, platform dependent sequencing errors and certain features of the reference genome, such as repetitive DNA sequences, pose challenges in this step. BWA-MEM [46] is a recent algorithm employed in BWA that is designed for longer reads and is faster and more accurate than previous methods. BWA-MEM (version 0.7.10) was used for aligning the quality filtered reads to the reference genome using default parameters. Toolboxes such as Bedtools [60] include various utilities

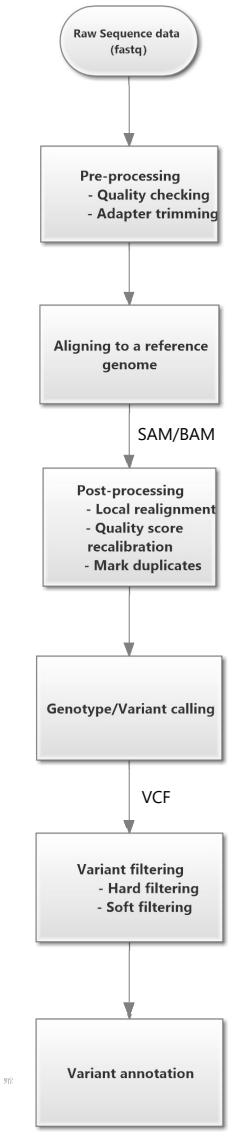


Figure 2: Variant/genotype calling workflow

for manipulating the alignment outputs (SAM/BAM files). The aligned read depth of the pools and the ENA runs was obtained from the alignment BAM files using Bedtools' coveragebed utility (version 2.17.0).

The next step, post-alignment processing deals with issues such as multi-reads (reads that aligned to multiple locations in the reference), PCR-artifacts (over-representation of certain sequences), inaccurate base qualities, alignment-inconsistencies (arising from independently aligning each read to the reference as opposed to multiple reads to reference mapping) and it is mostly application and type of sequencing data dependent. For instance, for studies that do not employ PCR amplification or for

which the removal of PCR duplicates has a negative consequence such as Pool-seq, removing PCR-artifacts could be skipped. The Picard tool (version 1.122) [2] was used to mark duplicates in the six individual strains and the ENA runs. Different aligners handle multi-reads differently; some discard all such reads while others choose one randomly if the alignments are equally best. Read re-alignment that considers the context of multiple reads, for example around INDELs, employed by certain variant calling tools can improve variant detection and genotyping accuracy [37]. For certain types of variant callers such as Freebayes and GATK's HaplotypeCaller, a separate re-alignment around INDELs step is not necessary as the tools already undergo local de novo assembly around areas that show evidence of variation. The Phred-scores reported by the base calling algorithms are very essential to the subsequent steps but they are often inaccurate [10]; base quality re-calibration aims to recalibrate the raw Phred-scores by using non-polymorphic sites so that they accurately reflect the true error rate. In GATK's implementation of the quality score recalibration, first the bases are grouped in to different categories based on the position of the base in the read, the raw quality score and the dinucleotide context. The quality score for each category is then estimated by using the number of mismatches against the reference genome; a comprehensive list of known or highly confident SNPs is required to infer the non-polymorphic sites. Re-alignment around INDELs and base quality re-calibration was undertaken using GATK before variant calling by UnifiedGenotyper. For the BaseRecalibrator, the variants consensually identified by Freebayes and the first run of UnifiedGenotyper were used since known variants are not available for GAS.

In the variant and genotype calling step, the base calls and their quality score will be used to identify positions that show evidence of variation compared to the reference and to assign genotypes (homo/heterozygous) to these variants. There is an uncertainty associated with variant calling due to errors, including those that occur during base calling and alignment, and variant calling tools need to account for that. Previous variant calling methods used hard cutoffs on per-base quality, total read depth, read alignment quality etc. to filter variants. For instance, in [73], six steps including the following thresholds, Phred quality score of 20, four supporting reads and an overall depth of 100 reads were used for filtering. These methods perform well when the sequencing depth is high but miss a lot of heterozygous calls when the depth is medium to low and also fail to quantify the uncertainty in the genotype inference

[58]. Currently, tools that use probabilistic methods such as Bayesian models are widely adopted. Through the Baye's formula, the posterior probability of a genotype G is inferred from the genotype likelihood, $P(Data|G)$, (which incorporates information such as the called base, per-base quality and alignment quality scores of the reads) and the prior probability of each genotype P(G), which can be set based on evidence from multiple samples (could incorporate LD information), or in single sample cases using external information such as variant databases. The genotype with the highest posterior probability will then be chosen as the most likely genotype, with its posterior probability or the ratio between its probability and the next highest used as a measure of uncertainty.

In such probabilistic frameworks, the ability to incorporate additional information from various sources results in more accurate genotype calling than the previous methods. The three tools utilized in this study, SAMtools [49], GATK's UnifiedGenotyper [23] and Freebayes [31], use such bayesian frameworks. There are also tools that use frequentist approaches such as SNver [74]. The Bcftools utility in SAMtools (version 1.1) was used to call variants from the pools, the six individual strains and the ENA runs after the *mpileup* command was used to calculate the genotype likelihoods. For the pools, besides SAMtools, GATK's UnifiedGenotyper (version 3.2-2), Freebayes (version 0.9.18-1) and SNVer (version 0.5.3) were used for calling variants. In SAMtools, maximum reads per input bam (–max-depth) of 10000 (default is 250) and a minimum mapping quality of 20 (default is 0) was utilized. In UnifiedGenotyper, a minimum Phred-scaled confidence threshold of 20 for calling and emitting variants was adopted. In Freebayes and SNVer, a variant was called if the minimum fraction of observations supporting the alternate allele was 0.02 and only bases of quality 13 or greater were counted (base quality of 13 is the default in SAMtools). The variants identified by the four tools were concatenated using bcftools *concat* command with the -d option to remove duplicates (version 1.2).

The variant calls are likely to contain false positives and in the filtering variant candidates step, we aim to improve the final call set by removing such artifacts; the filtering can be hard or soft. In hard filtering, a specific threshold is set on such things as variant call confidence scores, coverage of depth and mapping quality while in soft filtering an extensive variant database is used to learn the filtering criteria from the data itself instead. In the current study, a hard filtering of the variant

quality score threshold (20) was employed to filter variants from the pools, the six individual strains and the ENA runs.

In the last step, the annotation of the variants is undertaken to interpret the variants generated, for instance the effects of the variants on coding regions of the genome. SnpEff and SnpSift (version 4.0e) [16] were used to annotate and filter the variants that fall in the coding regions and had a non-synonymous effect.

Whole genome alignment and variant calling poses a different challenge than short read alignment due to the existence of non-linear rearrangements. C-Sibelia [53] handles this problem by breaking a genome in to synteny blocks which allows the separation of linear (SNPs, INDELs) and non-linear (rearrangements) operations; variants are then identified from the synteny and alignment blocks. The variants among the 20 complete genomes were identified by using C-Sibelia (version 3.0.5).

Different variant calling tools report the same variant sequence in different manners. This makes integration and comparison difficult across datasets. Vt [71] offers a way of normalizing the inconsistent variant representations. For instance the *decompose_blocksub* command will decompose the following representation of a multi-nucleotide polymorphism (MNP), CA/TG (REF/ALT) in to C/T and A/G. Vt was used to normalize the variant calls from the different tools before concatenation and comparison. Seqtk [45] is a lightweight tool that includes different functions to manipulate sequence data; it was used to sub sample reads from one of the pools.

3.5 Protein database creation

An inhouse python script was utilized for the database creation. Figure 3 illustrates the important steps of the process. First, the non synonymous variants that fall on the coding regions were chosen using the variant annotation tool Snpeff [16]. The ensemble of the non synonymous variants (SNPs and short INDELS) identified by all the four tools were incorporated in the database. The genbank format of the reference genome, that contains the sequences and the associated annotations, was downloaded from NCBI and used in the script. The nucleotide sequence of each protein was extracted from this file to insert the variants.

Since the pools were sequenced without indexing, it was impossible to determine

which variants came from which of the 50 strains that were in a pool. As a result, in our first attempt, we employed a combinatorial approach where we incorporated every possible combination of the variants in a protein and created separate entries in the database for each. However, such an approach resulted in very large databases and wasted a lot of computation time especially for proteins that contain large number of variants.

We then, adopted an alternative strategy of interrogating every read that mapped to a coding region and contained one or more of the final variant calls. This enabled us to determine and capture, to a certain degree, the strain specific variant signatures of a protein since more than one variable sequence could be included in the database for the protein if the reads were unique in their variant composition. The sam flags produced during alignment of the reads were also used for filtering reads, such as those that did not pass platform/vendor quality controls.

The variants were inserted in the protein sequences according to their positions. In the cases of INDELs, the original reported positions of the other variants that fell in the same protein had to be re-calculated based on how many nucleotides were inserted or deleted before them. On the other hand, if a read contained a mutation in the start codon resulting in a start loss, it was discarded. And, if a read contained frameshift mutations that caused premature stop codons or loss of stop codons, the protein sequence was truncated or elongated accordingly.

Following the inclusion of all the variants from a read into the nucleotide sequence of the protein (and reverse complementing it if it was on the complementary strand), *in silico* tryptic digestion of the protein took place; the peptides were cut when encountering an arginine (R) and lysine (K) residues unless a proline (P) follows immediately after. The peptide sequence that encloses the variants together with the two tryptic peptides that flank it, was then written as an independent entry in the fasta database. The flanking peptides were added to accommodate missed cleavage identifications.

In the fasta header, the protein accession id and information of the variant positions in that entry was recorded. To allow the concise representation of the variant information in the header, all the variant positions of a read were coded using bitwise flags (0 or 1) and then converted to decimal notation. For instance, if a protein

contains 4 variants in total and there is a read that contains the first and last variants, then the corresponding bitwise flag would be 1001 which in decimal form becomes 9.

Peptides less than 4 amino acids long were excluded (since they can not be positively identified in MS analysis), as were non unique peptides. Besides these variant peptides, the original sequences of these proteins as well as those proteins that had no variations were also included in the database, identified by 'org' and '0_org' headers respectively.

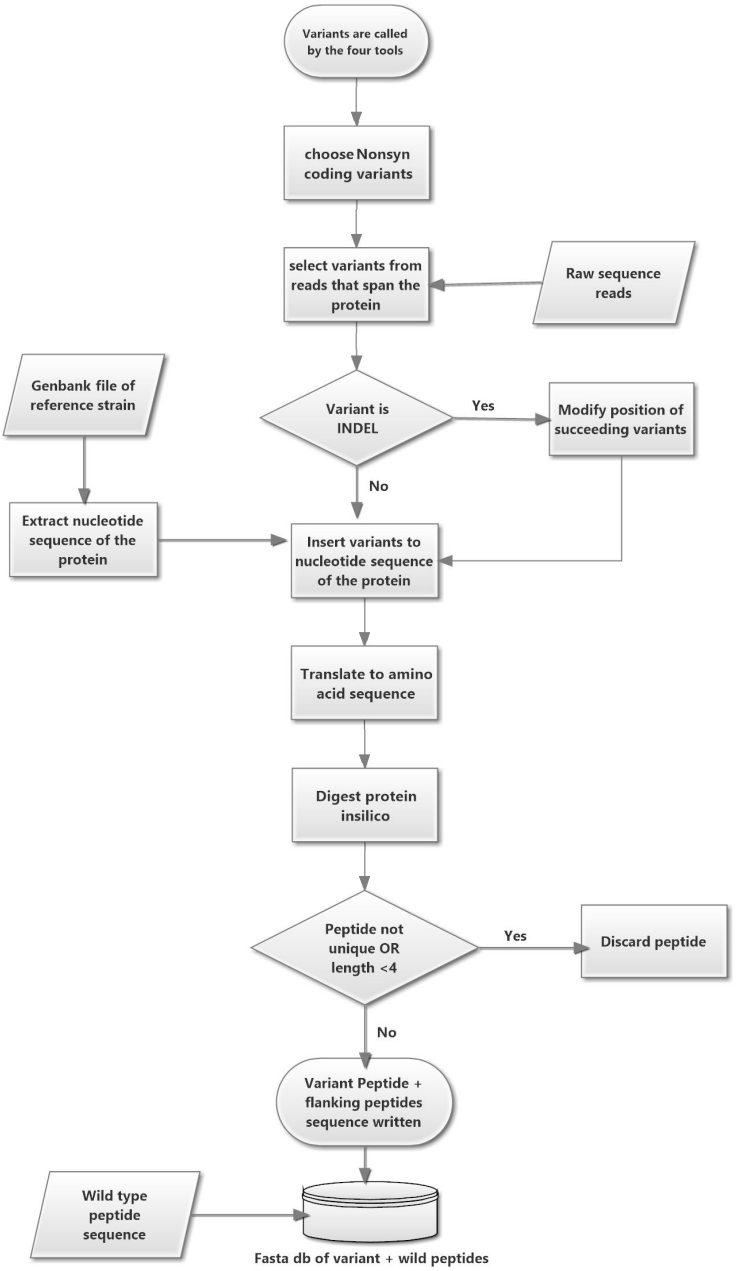


Figure 3: Flow chart of the database creation process

3.6 Shotgun tandem MS analysis

The Q Exactive™ Hybrid Quadrupole-Orbitrap™ Mass Spectrometer was coupled with nLC1000 liquid chromatography nanoflow system (ThermoScientific, USA) for the shotgun tandem MS analysis. The analysis was performed as described in appendix E.

4. Results

In this chapter, I discuss the results of the variant analysis from the pools and also the public data. The high sequencing depth and adequate pool size employed in this study enabled the identification of majority of the variants from the individual strains. Performance comparison of the four variant calling tools based on these identified variants is also included. In addition, preliminary results obtained by using the variant protein databases in an MS-based shotgun proteomics study of GAS is put forward.

4.1 Identification of genetic polymorphisms from the pools

High read depth and large pool sizes decrease the effect of the variance in DNA concentrations of the different samples. Most of the bases in our pools were spanned by larger number of reads than in the ENA data. Figure 4 shows, ∼90% of the bases in the two pools were covered by ∼ 10000 reads, while in ENA on average, they were covered only by ∼100 reads.

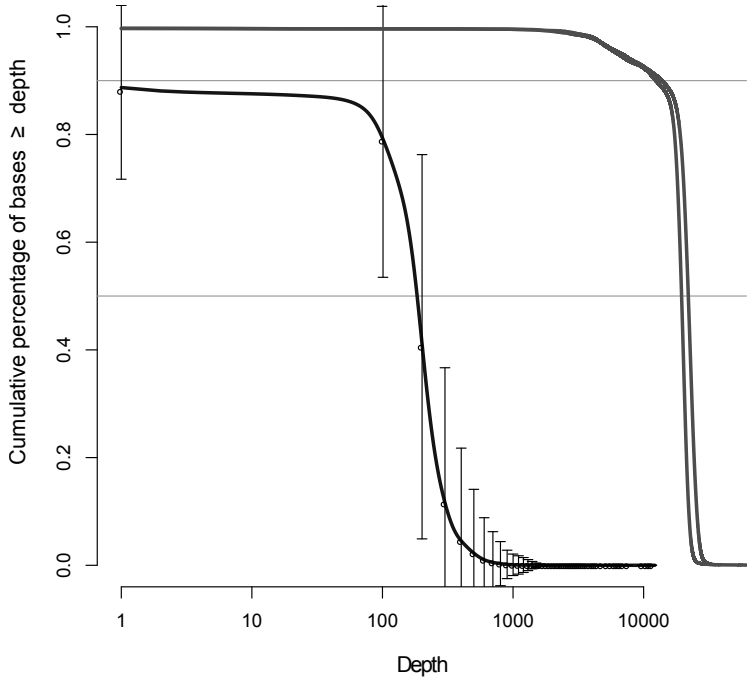


Figure 4: Cumulative percentage of bases spanned by at most X reads of the ENA runs and the pools. For the 3407 ENA runs the average is shown with ±1 SD. The coverage was calculated from the aligned reads. ∼90% of the bases in the two pools were covered by at most 10000 reads.

The two pools had average read depths of ∼18000x and 20000x (400x per strain) which is much higher than the average per strain read depth (∼186x) of the ENA data. Pool-seq has been effectively used for variant calling in various studies with much lower read depth than that of our study. For instance, Holt et al. [36] achieved ≥80% sensitivity at 40x. To study the effect of lower coverage in our study, we undertook a saturation analysis where we randomly sub sampled reads from one of our pools initially at average read depths of 10000x, 5000x, 1000x and 300x (and later added 4500x, 4000x, 3500x, 2500x, 2000x, 1500x). Following [3], we modeled the relationship between depth and number of variants using the Michaelis–Menten equation (Figure 5). At around 5000x, the number of variants identified was increasing only slightly as the coverage kept increasing.

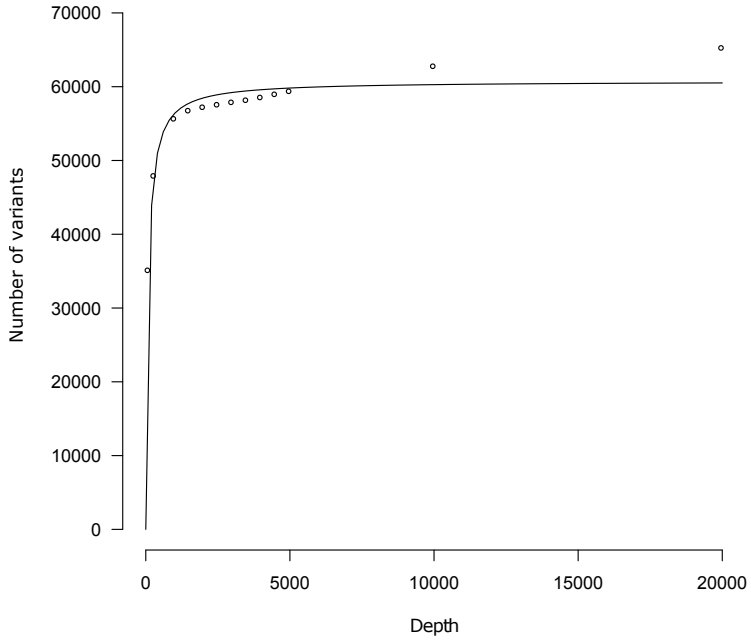


Figure 5: The number of variants identified (y-axis) at a given depth of coverage (x-axis). The 'drc' package in R was used to fit the data to the Michaelis–Menten equation.

A depiction of the read depth in 100 base stretches of the genome in Figures 6 and 7 reveals that regions bounding prophages had minimal coverage while areas, such as where a 23s ribosomal RNA resides, had higher coverage. Prophages are highly divergent structures responsible for the heterogeneity observed in different GAS strains while rRNAs are mostly conserved among species.

Variants were identified by using four tools, SAMtools, Freebayes, GATK's UnifiedGenotyper, and SNVer. The decision to use more than one tool was motivated by reports of varying concordance in calls by different tools [47, 59]. The concordance

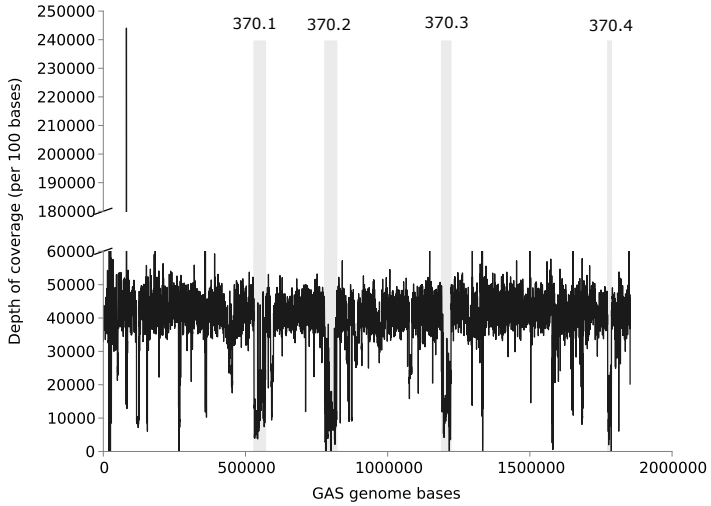


Figure 6: The aligned average coverage of the two pools per 100 base length of the GAS genome. The positions of prophages of the SF370 strain are indicated by the gray fill and the prophage numbers (370.1 etc.)

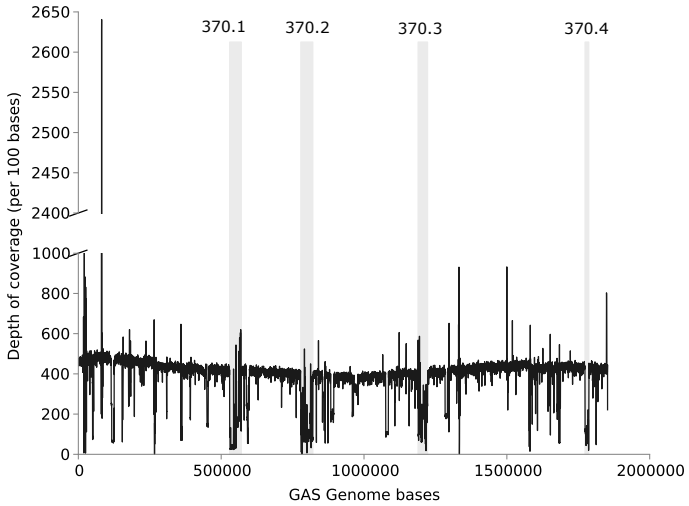


Figure 7: The aligned average coverage of the 3407 ENA runs per 100 base length of the GAS genome. The positions of prophages of the SF370 strain are indicated by the gray fill.

of SNPs and INDELs called by the four tools in this study endorses the findings from these studies (see Figure 8). The commands used to invoke these tools can be found in appendix C. The performance of each of the tools was evaluated by utilizing variants identified by SAMtools from six individually sequenced strains. Since SAMtools is the only tool that does not handle Pool-seq data, using it to call variants from the individual strains will help reduce bias while assessing the efficacy of the other tools.

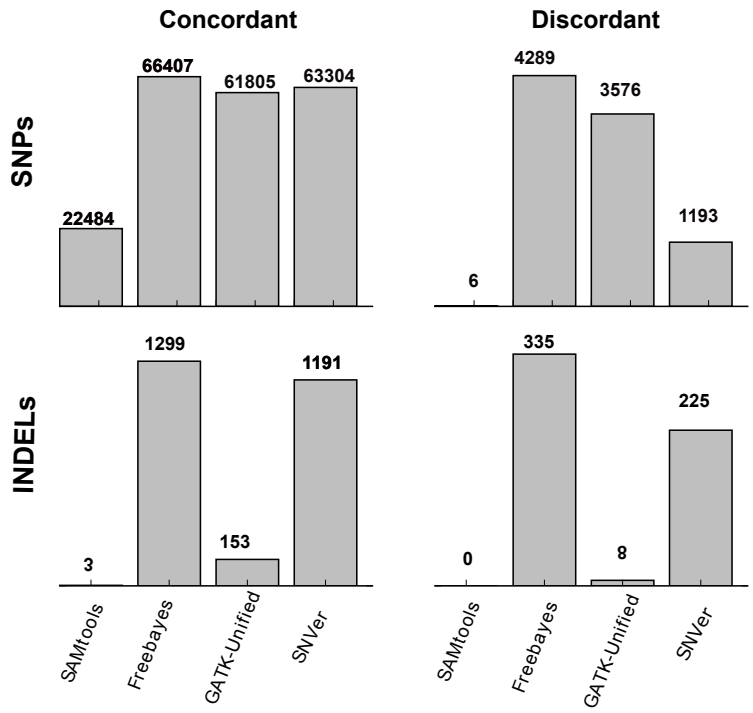


Figure 8: The concordant and discordant SNP and INDEL call counts of the 4 variant calling tools used in this study. The combined calls from the 4 tools is used as the true set for this analysis.

All tools, with the exception of SAMtools, identified more than 95% of the variants in all the individual strains (Table 1). The number of variants identified from the 6 individual strains increased by about 1% when calls from all the four tools were combined. To deter propagating errors from confounding evidence of true variation, it is a common practice to remove PCR duplicates before variant calling in studies that use PCR for amplification purposes; reads that have the same 5' position are considered duplicates. However, we anticipated removing such duplicates in Pool-seq studies will have a negative impact since reads that begin from the same position might come from different samples and are not PCR duplicates. As predicted, removing duplicates resulted in SAMtools failing to find 20% of the variants it identified before duplicate removal (see Table 1). *Gautier et al.* [32] removed duplicates in their Pool-seq study; however they used paired-end RAD sequencing whose random shearing mechanism makes it safe for removal of duplicates [22].

Using all the four tools without duplicate removal, 78955 variants were discovered from the two pools combined. Of these, 29% fell in coding regions and had non synonymous effect. Figures 9 and 10 show respectively, how the identified SNPs and INDELs are distributed across 100 base regions through out the GAS genome.

Table 1: Percentage of the variants identified from the six individual strains that were also mined from the pools using different variant calling tools and methods

	Individual strains					
Variant calling tools	**strain1**	**strain2**	**strain3**	**strain4**	**strain5**	**strain6**
Samtools (non-dedup)	67.2	71	71.2	69.9	63.4	70
Samtools (dedup)	48.1	52.2	50.9	47.6	47.4	51.4
Freebayes	96.9	97	96.9	96.8	95.7	96.5
GATK	95	94.9	94.9	94.9	92.2	94.3
SNVer	97.1	97	97	96.9	95.6	96.5
All combined	97.7	97.7	97.8	97.8	96.6	97.4

Except in some areas that showed very high and low variations, most of the variants identified had a uniform spread. Almost all the areas that had more than 30 SNPs per 100 bases were in and around putative genes whose function has yet not been determined in the reference strain. The area that contained the largest number of INDELs is where the scl (streptococcal collagen-like) gene that encodes a protein for attachment to the host epithelial cells resides.

We also analyzed the polymorphisms of one of the pools using the other 19 genomes as a reference to determine how the choice of the reference genome will impact the variant analysis. The Number of SNPs and INDELS from this analysis ranged from 62680 to 72133 and 1271 to 1551 respectively (Figures 11 and12).

4.2 Identification of genetic polymorphisms from public data

The 19 publicly available complete genomes were aligned to the reference genome and the variants identified from each were merged. In total, 62600 SNPS and 1469

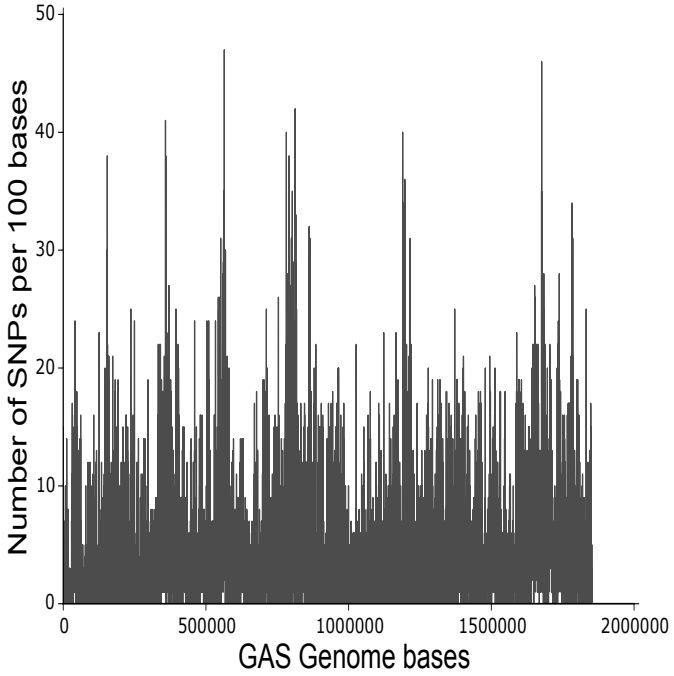


Figure 9: Distribution of the SNPs identified from the pools per 100 base length. The SNPs are mostly uniformly distributed across the reference genome

Table 2: Total number of variants identified in the two pools, the 19 complete genomes and 3407 runs from ENA

	Two pools	**19 complete GAS genomes**	**3407 ENA runs**
Total number of variants	78955	65334	286502
SNPs	76981	62600	270212
INDELs	1725	1469	16290

INDELs were discovered (see Table 2), which as in the pools were more or less uniformly distributed (data not shown). We further compared the 20 genomes with one another and clustered them based on the number of variants among them, which ranged from 100 to 9286 (Figure 13). A table containing this pairwise variant counts is attached in appendix B.

From ENA, initially 3513 paired end illumina sequencing runs that had varying

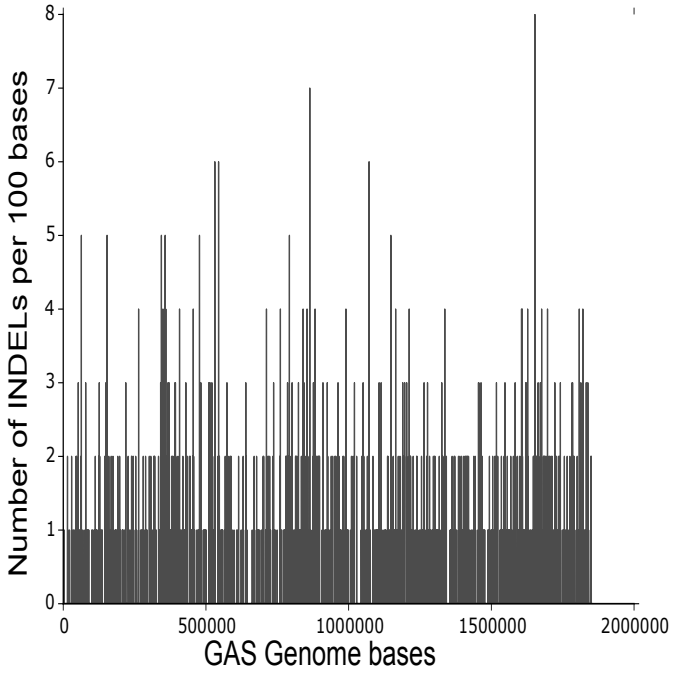


Figure 10: Distribution of the INDELs identified from the pools per 100 base length. The INDELS are mostly uniformly distributed across the reference genome

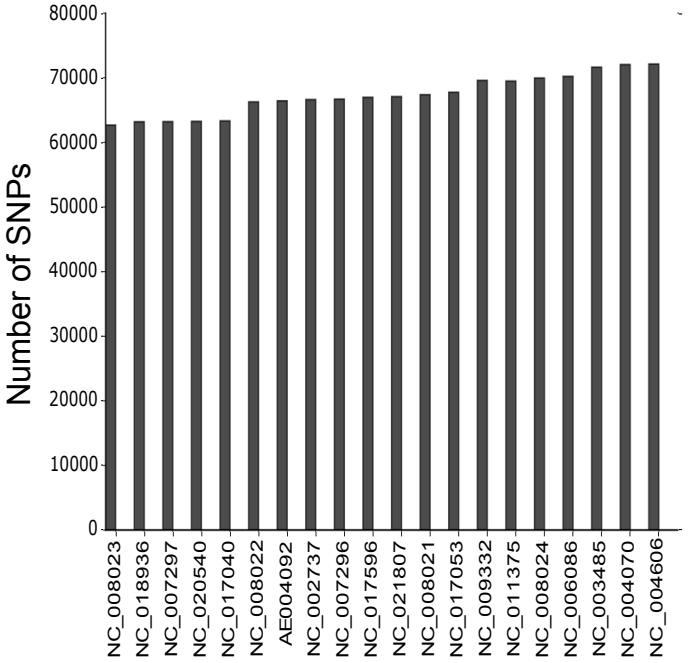


Figure 11: SNPs identified from one of the pools relative to the 20 complete GAS genomes publicly available

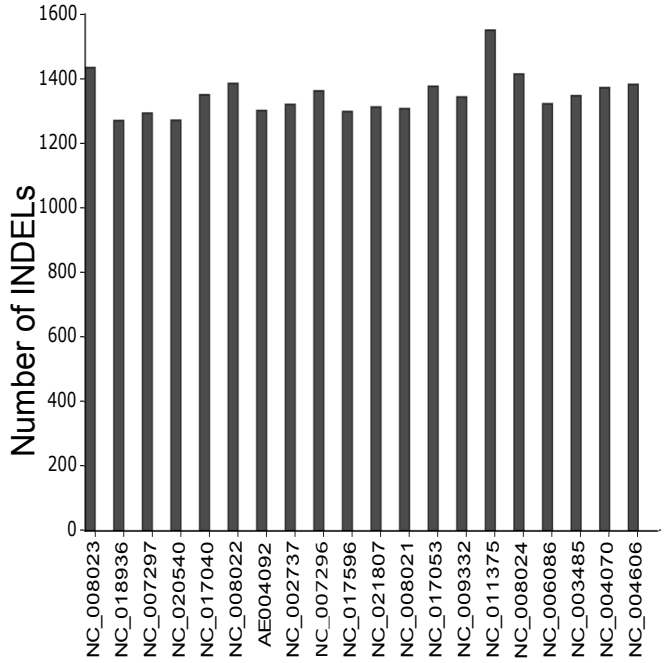


Figure 12: INDELs identified from one of the pools relative to the 20 complete GAS genomes publicly available

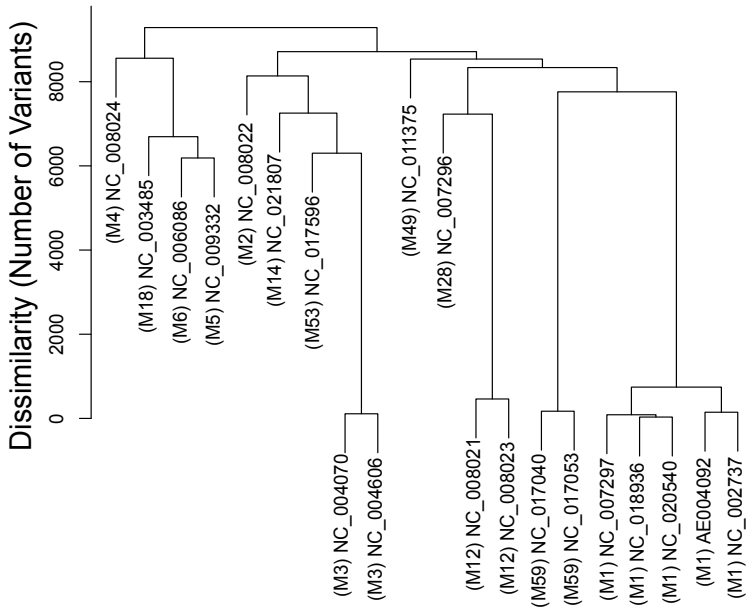


Figure 13: Hierarchical clustering of the 20 publicly available GAS genomes based on the number of variants that exist among them. The M-types of each of the strains is given in parenthesis

degrees of sequence quality (from 1% to 100% mappability to the reference genome) were aligned to the reference. There are a variety of uncertainties associated with publicly available data. One such uncertainty concerns the species nomenclatures

used. For instance, it is not unusual to find out that sequences deposited as GAS are actually not, therefore the need for filtering. In addition, according to a personal correspondence with the support team at EMBL-EBI, the same experimental runs could be assigned more than one id. This is the reason why we are referring to the data from ENA as runs and not genomes.

We removed 106 of the ENA runs that had mapping percentage of less than 1.5*IQR (a mapping percentage of $< 83.93\%$) from our variant analysis. After the exclusion of such runs, the 3407 runs that remained had an average mapping and proper pair percentages of 93.3 and 91.01 respectively (Table 4). There were 286502 variants from the 3407 runs and they contained 95.7% and 90.6% of the variants identified in the two pools and the 19 genomes respectively (Table 3).

Table 3: Percentage of variants identified from the two pools, 19 GAS genomes and the ENA runs (rows) that were also found in two of these (columns)

	Two pools	**19 complete GAS genomes**	**3407 ENA runs**
Two pools		60.5	95.7
19 GAS genomes	67.9		90.6
3407 ENA runs	24.8	20.9	

To investigate if certain regions were more variable in the pools than the 19 genomes or the ENA data, we divided the entire genome of the reference strain into 10 kb regions and calculated the proportion of variants in such regions (Figure 14). Some areas appeared to have different proportions in these three sets; Fisher's exact test was used to identify those regions that show significant differences. There were two areas in particular where the ENA data showed the highest difference compared to the pools and also the 19 genomes; these were locations of the 370.1 and 370.2 prophages of the SF370 genome (Figure 15). Even though there were such areas that had differences in the proportion of variants, in general, based on the welch-ANOVA test, the means of the proportions of the three sets are not statistically different (p-value of 0.997).

Table 4: Alignment statistics of sequence reads from the two pools and the 3407 ENA runs. For the ENA runs, besides averages, maximum and minimum values are given for the mapped, unmapped and properly paired percentages

	Two pools	**3407 ENA runs**
Total number of reads	$8.0 * 10^8$	$1.4 * 10^{10}$
Mapped (%)	91.7	93.13 (83.9-100)
Unmapped (%)	8.2	6.8 (0-16)
Properly paired (%)	86.9	91.0 (78.0-99.6)

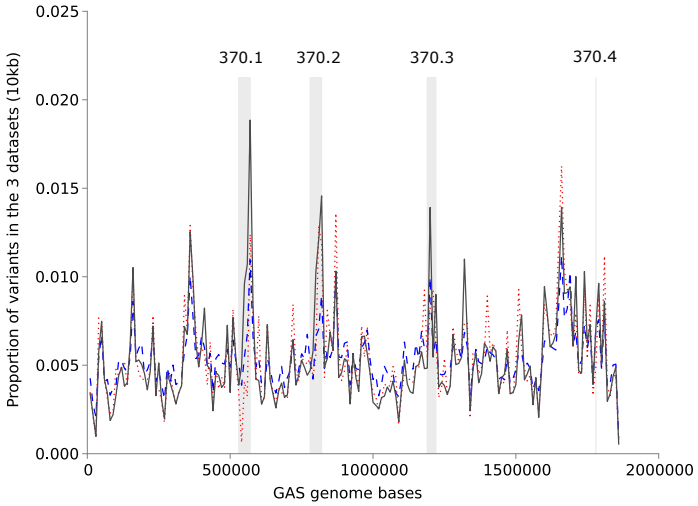


Figure 14: Graph showing the relative variability of 10 kb regions. Black solid lines represent the pools, red dotted lines the 19 genomes and blue dashed lines the ENA data. The positions of prophages of the SF370 strain are shown with gray fills.

4.3 Variant protein databases

For the shotgun proteomics experiment, first, 19429 of the variants (from the total of 76981 SNVs and 1725 INDELs identified by the four variant calling tools) that fall within the protein coding regions and that had non-synonymous effects were selected. Using the in-house script these variants were then incorporated to the respective proteins of the reference genome. Afterwards, the protein sequences were *in silico* digested and both wild and variant peptides were written to the fasta database.

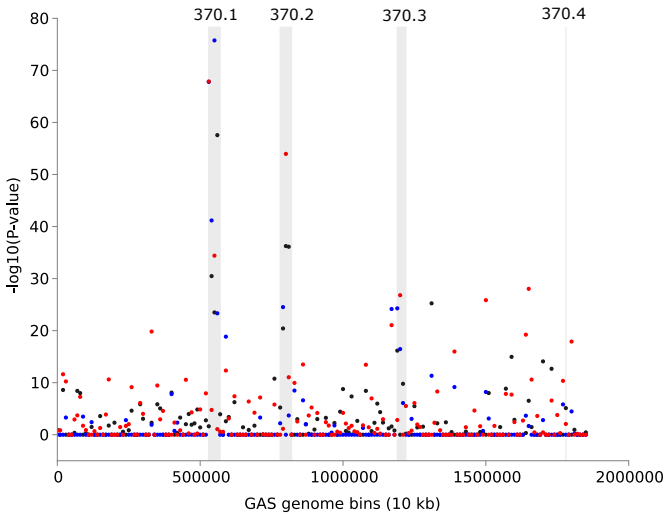


Figure 15: -log10 p-value from a fisher's exact test to identify regions that show significant difference in the proportion of variants between the pools, the 19 genomes and the ENA data. Black dots indicate the p-values of the proportions of variants between the pools and the ENA data, blue dots the p-values between the pools and the 19 genomes and red dots the p-values between the ENA data and the 19 genomes. The positions of the prophages of the SF370 strain are shown with gray fills.

The inclusion of the variant peptides brought about an increase of 8.5% in the database size, which is almost double the increase reported by *Li et.al* [50]. This could be attributed to the read based strategy employed in this study to account for the pooled nature of the data. There were ∼68916 variant peptide entries from 1697 proteins in each of the databases.

The variant peptide database was used to search for spectra matches from 5 GAS strains using the PEAKS DB search engine. Different protein extraction methods were employed to target proteins that are found in different subcellular locations.

Table 5, lists the peptide matches for one such preparation, the trichloroacetic acid (TCA) method, that targeted the surface proteins. On average, ∼ 200 variant peptides were identified from each of the 5 samples. Figure 16 shows that the proteins that matched to wild type peptides have less number of variants per amino acid than those that matched to the variant peptides. This suggests that the highly variable proteins may not find a match in conventional databases that do not contain variants. However, some of these variant peptide matches could also be false positives, found as hits only because of the increase in the database size; in Figure 17 more of the variant peptides have scores shifted to the lower end than the wild peptides (see also Fig 2a in [50]).

Table 5: Number of wild and variant peptides identified in the TCA preparation from 5 GAS strains

Samples	Wild peptides	Variant peptides
161072	2713	263
252285	935	126
253082	1549	169
253411	2422	258
253414	2393	277

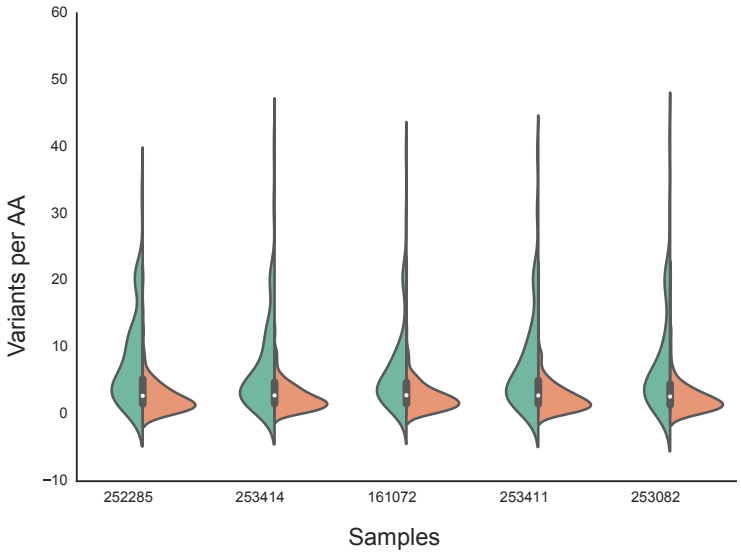


Figure 16: Distribution of the number of variants for proteins that matched to the variant (green) and wild type peptides (orange). Proteins that could be identified with the variant peptides had higher number of variants than those that matched to the wild type peptides

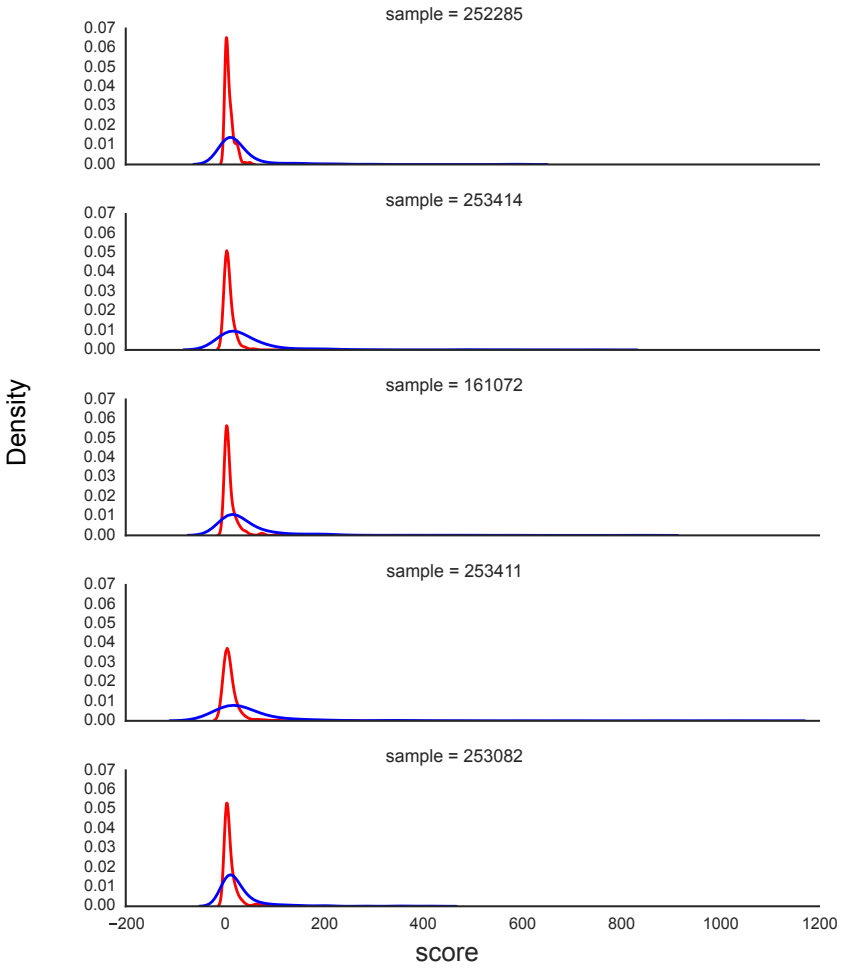


Figure 17: Distribution of search scores for the variant (red) and wild type peptides (blue). More variant peptides than wild ones are in the low range.

5. Summary

Bacteria are diverse and can thrive in extreme conditions. Some of them affect health and thus the economy. For certain species, the intra-species diversity is very high that the available sequenced strains might not be representatives, requiring the sequencing of large number of strains.

The arrival of NGS technologies has facilitated the sequencing of various non model organisms. Moreover, technological advances in proteomics such as in mass spectrometry and the substantial and ever growing availability of data in the public domain has promoted interfacing between genomics and proteomics. In GAS, various strains of both invasive and non-invasive nature have been whole genome sequenced. But, undertaking studies that require sequencing of large number of individuals is still expensive for many research groups, necessitating the need for cost effective alternatives such as Pool-seq.

In this study, we demonstrated Pool-seq to be an efficient alternative for genome-wide polymorphism surveys given high enough sequencing coverage and the right choice of variant calling tools and methods. We utilized the variants mined using this approach to develop a custom database for GAS to be used in MS based proteomics.

In Pool-seq, the precision of variant detection increases with the increase in coverage and decreases with the minimum number of reads required for allele calling [30]. In this study, at the average read depth of ∼20000x, we were able to detect most of the variants in the six individually sequenced strains with the false negative rate being at most ∼4%, i.e. given that we consider all the variants identified from the individual strains as true variants, which usually is not the case. The variants Pool-seq failed to uncover had a small average raw depth (in the tens range) compared to those that could be identified, which ranged in the tens of thousands. This indicates, similar to other studies [33, 35], that the approach is not suited for studies that rely on rare variants.

The resampling analysis also shows that Pool-seq is still a powerful approach for variant discovery even at a coverage that is four folds smaller than the average read depth taken up in this study. We note that we used only few individually sequenced strains for verifying the efficiency of Pool-seq and as a result the false negative rate

could be higher for some strains due to differential representation in the pool. Further, because we had variant information for only six of the 50 strains in a pool, we cannot report the false positive rate from our study.

Various studies have demonstrated the effect the size of the pool has on the performance of Pool-seq. For instance, *Anderson et al.* [5] in their commentary presented how small sample sizes and low read depth led to an erroneous inference of pronounced population structure in Baltic sea herring data published by *Corander et. al* [17]. A minimum pool size of 40 is recommended in Pool-seq studies.

In our comparison of variant calling tools, using SAMtools with duplicate removal resulted in the identification of the least number of variants from the six individual strains. We have come across Pool-seq studies that used SAMtools and/or removed duplicates. *Mullen et al.* [54] used SAMtools for variant detection and removed PCR duplicates which resulted in identification of only 598 of the 1304 (45.9%) dbSNPs variants. They attributed the high false negative rate to low coverage, lack of segregation of the SNPs and inaccurate dbSNP data. However, based on results from our study (on average only 49.6% variants from the 6 strains were identified; see Table 1), we think it is likely that the choice of SAMtools for variant calling and duplicate removal contributed to such a high false negative rate. We therefore recommend, at least for whole genome Pool-seq studies with high coverage, to use variant calling tools that can handle pooled samples and to avoid duplicate removal.

On the other hand, *Gautier et al.* [32] observed for their RAD data that without duplicate removal the "effective pool size" substantially decreased which led them to conclude that PCR duplicates considerably contribute to the overall experimental error. Hence, in certain Pool-seq studies, for instance those that use RAD sequencing, duplicate removal might be appropriate. Furthermore, there are other methods for handling duplicates that consider other base qualities besides read positions, such as that employed by *Chen et al.* [14], which might be more appropriate under certain circumstances.

Having access to publicly available sequences of various GAS strains proved essential in our study. We used the public data to validate the variants we identified from the pools. We have also acquired valuable insights about GAS in general including variations that exist in various strains. For instance, we have identified a 23rRNA

that is highly conserved among different strains of GAS; a blast search confirmed that the rRNA is also conserved in other streptococcal groups such as groups B, C and G. We also found the phage regions of the reference strain to have the lowest coverage, in the public ENA data as in our pools, which lends further support to previous studies on the highly variable nature of phage and phage-like elements (see [69, 6]).

We were also able to cluster the 20 publicly available GAS strains based on the number of variants between them and found M59 strains to be more similar to the M1 strains than strains of other M serotypes. In our analysis of the public data we also found that only small percentages of the variants from the ENA data were discovered in our two pools and the 19 complete genomes. This shows that the strains in our pools and the complete genomes represent only but a small proportion of, geographically limited, strains and more can be uncovered about GAS through the sequencing and annotation of additional strains.

In general, many studies involving non-model organisms can benefit from sequencing of large number of individuals and Pool-seq can be used in such cases when the cost and effort of sequencing individuals is unaffordable. In this study, we have examined issues such as the expected coverage, representation of samples in a pool and variant calling methods that are central to a Pool-seq genome wide polymorphism detection study. We have demonstrated that Pool-seq can be an efficient cost effective alternative in polymorphism discovery for large samples of organisms that already have a high quality reference genome.

On another note, novel peptides can be identified in shotgun proteomics studies by searching against customized protein databases that contain genomic and proteomic sequence information. For instance such databases have been generated from RNA-seq data [76], genomic variant data [50] and EST data [25]. Bacteria, such as GAS, that exhibit very high variability could benefit from the availability of variant protein databases. The preliminary results obtained from this study suggest that identification of peptides/proteins can be greatly improved by employing such custom databases. However, the various issues, such as the database size and false peptide identifications, associated with such databases need careful considerations [56].

5.1 Future Improvements

In this study, around 10% of the sequence reads could not be aligned to the reference genomes. Investigating these reads further will be part of our future work. Since the typical way of removing duplicates using Picard or SAMtools resulted in lower sensitivity in our Pool-seq analysis, we have not removed duplicates in this study. However, the saturation analysis of the sequencing depth (versus number of variants) revealed we could have achieved a comparable sensitivity at a depth of coverage that is 4 folds smaller and we anticipate the high depth of coverage could lead to higher number of false positives. We therefore plan to remove duplicates using other methods such as those that take in to consideration additional information besides the 5' end position [14] or by using binomial distribution to calculate the maximum number of reads at a certain position as in [78]. We also plan to identify markers for distinguishing invasive and non-invasive strains by employing the insights from the genomics and proteomics analysis that was carried out in this study. We hope the availability of the custom databases will facilitate such future explorations.

From the analysis of the publicly available sequence data, it was evident that there are but much larger number of variants than that we could identify from our pools. We therefore plan to improve our protein databases by including these variants. In addition, previous studies have shown that there is a high risk of false positives associated with such databases and therefore separate false discovery rates (FDRs) need to be specified for the variant and wild type sequences. Hence we plan to modify our current search settings that use the same FDR threshold of 0.05 for both variant and wild type peptide matches and instead employ the method of *Li et.al* [51] or a modified version that will be more suitable for pooled data.

References

[1] Novocraft. http://www.novocraft.com/products/novoalign/. Accessed: 2016-04-07.

[2] Picard. http://broadinstitute.github.io/picard/. Accessed: 2016-03-07.

[3] Alioto, T. S., Buchhalter, I., Derdak, S., Hutter, B., Eldridge, M. D., Hovig, E., Heisler, L. E., Beck, T. A., Simpson, J. T., Tonon, L., et al. A comprehensive assessment of somatic mutation detection in cancer using whole-genome sequencing. *Nature communications 6* (2015).

[4] Altmann, A., Weber, P., Quast, C., Rex-Haffner, M., Binder, E. B., and Müller-Myhsok, B. vipr: variant identification in pooled dna using r. *Bioinformatics 27*, 13 (2011), i77–i84.

[5] Anderson, E. C., Skaug, H. J., and Barshis, D. J. Next-generation sequencing for molecular ecology: a caveat regarding pooled samples. *Molecular ecology 23*, 3 (2014), 502–512.

[6] Banks, D. J., Beres, S. B., and Musser, J. M. The fundamental contribution of phages to gas evolution, genome diversification and strain emergence. *Trends in microbiology 10*, 11 (2002), 515–521.

[7] Bansal, V. A statistical method for the detection of variants from next-generation resequencing of dna pools. *Bioinformatics 26*, 12 (2010), i318–i324.

[8] Bentley, D. R., Balasubramanian, S., Swerdlow, H. P., Smith, G. P., Milton, J., Brown, C. G., Hall, K. P., Evers, D. J., Barnes, C. L., Bignell, H. R., et al. Accurate whole human genome sequencing using reversible terminator chemistry. *nature 456*, 7218 (2008), 53–59.

[9] Bolger, A. M., Lohse, M., and Usadel, B. Trimmomatic: a flexible trimmer for illumina sequence data. *Bioinformatics* (2014), btu170.

[10] Brockman, W., Alvarez, P., Young, S., Garber, M., Giannoukos, G., Lee, W. L., Russ, C., Lander, E. S., Nusbaum, C., and Jaffe, D. B. Quality scores and snp detection in sequencing-by-synthesis systems. *Genome research 18*, 5 (2008), 763–770.

[11] Bunger, M. K., Cargile, B. J., Sevinsky, J. R., Deyanova, E., Yates, N. A., Hendrickson, R. C., and Stephenson, J. L. Detection and validation of non-synonymous coding snps from orthogonal analysis of shotgun proteomics data. *Journal of proteome research 6*, 6 (2007), 2331–2340.

[12] Burrows, M., and Wheeler, D. A block-sorting lossless data compression algorithm. In *DIGITAL SRC RESEARCH REPORT* (1994), Citeseer.

[13] Carapetis, J. R., Steer, A. C., Mulholland, E. K., and Weber, M. The global burden of group a streptococcal diseases. *The Lancet infectious diseases 5*, 11 (2005), 685–694.

[14] Chen, X., Listman, J. B., Slack, F. J., Gelernter, J., and Zhao, H. Biases and errors on allele frequency estimation and disease association tests of next-generation sequencing of pooled samples. *Genetic epidemiology 36*, 6 (2012), 549–560.

[15] Chewapreecha, C., Marttinen, P., Croucher, N. J., Salter, S. J., Harris, S. R., Mather, A. E., Hanage, W. P., Goldblatt, D., Nosten, F. H., Turner, C., et al. Comprehensive identification of single nucleotide polymorphisms associated with beta-lactam resistance within pneumococcal mosaic genes. *PLoS Genet 10*, 8 (2014), e1004547.

[16] Cingolani, P., Platts, A., Wang, L. L., Coon, M., Nguyen, T., Wang, L., Land, S. J., Lu, X., and Ruden, D. M. A program for annotating and predicting the effects of single nucleotide polymorphisms, snpeff: Snps in the genome of drosophila melanogaster strain w1118; iso-2; iso-3. *Fly 6*, 2 (2012), 80–92.

[17] Corander, J., Majander, K. K., Cheng, L., and Merilä, J. High degree of cryptic population differentiation in the baltic sea herring clupea harengus. *Molecular Ecology 22*, 11 (2013), 2931–2940.

[18] Craig, R., and Beavis, R. C. Tandem: matching proteins with tandem mass spectra. *Bioinformatics 20*, 9 (2004), 1466–1467.

[19] Creasy, D. M., and Cottrell, J. S. Error tolerant searching of uninterpreted tandem mass spectrometry data. *Proteomics 2*, 10 (2002), 1426–1434.

[20] Cutler, D. J., and Jensen, J. D. To pool, or not to pool? *Genetics 186*, 1 (2010), 41–43.

[21] Dasari, S., Chambers, M. C., Slebos, R. J., Zimmerman, L. J., Ham, A.-J. L., and Tabb, D. L. Tagrecon: high-throughput mutation identification through sequence tagging. *Journal of proteome research 9*, 4 (2010), 1716–1726.

[22] Davey, J. W., Hohenlohe, P. A., Etter, P. D., Boone, J. Q., Catchen, J. M., and Blaxter, M. L. Genome-wide genetic marker discovery and genotyping using next-generation sequencing. *Nature Reviews Genetics 12*, 7 (2011), 499–510.

[23] DePristo, M. A., Banks, E., Poplin, R., Garimella, K. V., Maguire, J. R., Hartl, C., Philippakis, A. A., Del Angel, G., Rivas, M. A., Hanna, M., et al. A framework for variation discovery and genotyping using next-generation dna sequencing data. *Nature genetics 43*, 5 (2011), 491–498.

[24] Druley, T. E., Vallania, F. L., Wegner, D. J., Varley, K. E., Knowles, O. L., Bonds, J. A., Robison, S. W., Doniger, S. W., Hamvas, A., Cole, F. S., et al. Quantification of rare allelic variants from pooled genomic dna. *Nature methods 6*, 4 (2009), 263.

[25] Edwards, N. J. Novel peptide identification from tandem mass spectra using ests and sequence database compression. *Molecular systems biology 3*, 1 (2007), 102.

[26] Ellegren, H. Genome sequencing and population genomics in non-model organisms. *Trends in ecology & evolution 29*, 1 (2014), 51–63.

[27] Ferretti, L., Ramos-Onsins, S. E., and Pérez-Enciso, M. Population genomics from pool sequencing. *Molecular ecology 22*, 22 (2013), 5561–5576.

[28] Fischer, M. C., Rellstab, C., Tedder, A., Zoller, S., Gugerli, F., Shimizu, K. K., Holderegger, R., and Widmer, A. Population genomic footprints of selection and associations with climate in natural populations of arabidopsis halleri from the alps. *Molecular ecology 22*, 22 (2013), 5594–5607.

[29] Fleischmann, R. D., Adams, M. D., White, O., Clayton, R. A., Kirkness, E. F., Kerlavage, A. R., Bult, C. J., Tomb, J.-F., Dougherty,

B. A., Merrick, J. M., et al. Whole-genome random sequencing and assembly of haemophilus influenzae rd. *Science 269*, 5223 (1995), 496–512.

[30] Futschik, A., and Schlötterer, C. The next generation of molecular markers from massively parallel sequencing of pooled dna samples. *Genetics 186*, 1 (2010), 207–218.

[31] Garrison, E., and Marth, G. Haplotype-based variant detection from short-read sequencing. *arXiv preprint arXiv:1207.3907* (2012).

[32] Gautier, M., Foucaud, J., Gharbi, K., Cézard, T., Galan, M., Loiseau, A., Thomson, M., Pudlo, P., Kerdelhué, C., and Estoup, A. Estimation of population allele frequencies from next-generation sequencing data: pool-versus individual-based genotyping. *Molecular Ecology 22*, 14 (2013), 3766–3779.

[33] Guo, Y., Samuels, D. C., Li, J., Clark, T., Li, C.-I., and Shyr, Y. Evaluation of allele frequency estimation using pooled sequencing data simulation. *The Scientific World Journal 2013* (2013).

[34] Guzmán, E., Romeu, A., and Garcia-Vallve, S. Completely sequenced genomes of pathogenic bacteria: A review. *Enfermedades infecciosas y microbiología clínica 26*, 2 (2008), 88–98.

[35] Harakalova, M., Nijman, I. J., Medic, J., Mokry, M., Renkens, I., Blankensteijn, J. D., Kloosterman, W., Baas, A. F., and Cuppen, E. Genomic dna pooling strategy for next-generation sequencing-based rare variant discovery in abdominal aortic aneurysm regions of interest—challenges and limitations. *Journal of cardiovascular translational research 4*, 3 (2011), 271–280.

[36] Holt, K. E., Teo, Y. Y., Li, H., Nair, S., Dougan, G., Wain, J., and Parkhill, J. Detecting snps and estimating allele frequencies in clonal bacterial populations by sequencing pooled dna. *Bioinformatics 25*, 16 (2009), 2074–2075.

[37] Homer, N., and Nelson, S. F. Improved variant discovery through local re-alignment of short-read next-generation sequencing data using srma. *Genome biology 11*, 10 (2010), 1.

[38] Hong, S. N., Park, C., Park, S. J., Lee, C. K., Ye, B. D., Kim, Y. S., Lee, S., Chae, J., Kim, J.-I., Kim, Y.-H., et al. Deep resequencing of 131 crohn's disease associated genes in pooled dna confirmed three reported variants and identified eight novel variants. *Gut* (2015), gutjnl–2014.

[39] Hutchison, C. A. Dna sequencing: bench to bedside and beyond. *Nucleic acids research 35*, 18 (2007), 6227–6237.

[40] Jin, S. C., Pastor, P., Cooper, B., Cervantes, S., Benitez, B. A., Razquin, C., Goate, A., Cruchaga, C., et al. Pooled-dna sequencing identifies novel causative variants in psen1, grn and mapt in a clinical early-onset and familial alzheimer's disease ibero-american cohort. *Alzheimers Res Ther 4*, 4 (2012), 34.

[41] JP, E. List of prokaryotic names with standing in nomenclature—genus streptococcus. http://www.bacterio.cict.fr/s/streptococcus.html. Accessed: 2016-03-07.

[42] Koboldt, D. C., Zhang, Q., Larson, D. E., Shen, D., McLellan, M. D., Lin, L., Miller, C. A., Mardis, E. R., Ding, L., and Wilson, R. K. Varscan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing. *Genome research 22*, 3 (2012), 568–576.

[43] Land, M., Hauser, L., Jun, S.-R., Nookaew, I., Leuze, M. R., Ahn, T.-H., Karpinets, T., Lund, O., Kora, G., Wassenaar, T., et al. Insights from 20 years of bacterial genome sequencing. *Functional & integrative genomics 15*, 2 (2015), 141–161.

[44] Langmead, B., and Salzberg, S. L. Fast gapped-read alignment with bowtie 2. *Nature methods 9*, 4 (2012), 357–359.

[45] Li, H. Seqtk. https://github.com/lh3/seqtk. Accessed: 2016-03-07.

[46] Li, H. Aligning sequence reads, clone sequences and assembly contigs with bwa-mem. *arXiv preprint arXiv:1303.3997* (2013).

[47] Li, H. Toward better understanding of artifacts in variant calling from high-coverage samples. *Bioinformatics 30*, 20 (2014), 2843–2851.

[48] Li, H., and Durbin, R. Fast and accurate short read alignment with burrows–wheeler transform. *Bioinformatics 25*, 14 (2009), 1754–1760.

[49] Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R., et al. The sequence alignment/map format and samtools. *Bioinformatics 25*, 16 (2009), 2078–2079.

[50] Li, J., Duncan, D. T., and Zhang, B. Canprovar: a human cancer proteome variation database. *Human mutation 31*, 3 (2010), 219–228.

[51] Li, J., Su, Z., Ma, Z.-Q., Slebos, R. J., Halvey, P., Tabb, D. L., Liebler, D. C., Pao, W., and Zhang, B. A bioinformatics workflow for variant peptide detection in shotgun proteomics. *Molecular & Cellular Proteomics 10*, 5 (2011), M110–006536.

[52] Lunter, G., and Goodson, M. Stampy: a statistical algorithm for sensitive and fast mapping of illumina sequence reads. *Genome research 21*, 6 (2011), 936–939.

[53] Minkin, I., Pham, H., Starostina, E., Vyahhi, N., and Pham, S. C-sibelia: an easy-to-use and highly accurate tool for bacterial genome comparison. *F1000Research 2* (2013).

[54] Mullen, M. P., Creevey, C. J., Berry, D. P., McCabe, M. S., Magee, D. A., Howard, D. J., Killeen, A. P., Park, S. D., McGettigan, P. A., Lucy, M. C., et al. Polymorphism discovery and allele frequency estimation using high-throughput dna sequencing of target-enriched pooled dna samples. *BMC genomics 13*, 1 (2012), 16.

[55] Nasser, W., Beres, S. B., Olsen, R. J., Dean, M. A., Rice, K. A., Long, S. W., Kristinsson, K. G., Gottfredsson, M., Vuopio, J., Raisanen, K., et al. Evolutionary pathway to increased virulence and epidemic group a streptococcus disease derived from 3,615 genome sequences. *Proceedings of the National Academy of Sciences 111*, 17 (2014), E1768–E1776.

[56] Nesvizhskii, A. I. Proteogenomics: concepts, applications and computational strategies. *Nature methods 11*, 11 (2014), 1114–1125.

[57] Nesvizhskii, A. I., Roos, F. F., Grossmann, J., Vogelzang, M., Eddes, J. S., Gruissem, W., Baginsky, S., and Aebersold, R. Dynamic spectrum quality assessment and iterative computational analysis of shotgun proteomic data toward more efficient identification of post-translational modifications, sequence polymorphisms, and novel peptides. *Molecular & Cellular Proteomics 5*, 4 (2006), 652–670.

[58] Nielsen, R., Paul, J. S., Albrechtsen, A., and Song, Y. S. Genotype and snp calling from next-generation sequencing data. *Nature Reviews Genetics 12*, 6 (2011), 443–451.

[59] O'Rawe, J., Jiang, T., Sun, G., Wu, Y., Wang, W., Hu, J., Bodily, P., Tian, L., Hakonarson, H., Johnson, W. E., et al. Low concordance of multiple variant-calling pipelines: practical implications for exome and genome sequencing. *Genome med 5*, 3 (2013), 28.

[60] Quinlan, A. R., and Hall, I. M. Bedtools: a flexible suite of utilities for comparing genomic features. *Bioinformatics 26*, 6 (2010), 841–842.

[61] Rantala, S., Vuopio-Varkila, J., Vuento, R., Huhtala, H., and Syrjänen, J. Clinical presentations and epidemiology of β-haemolytic streptococcal bacteraemia: a population-based study. *Clinical Microbiology and Infection 15*, 3 (2009), 286–288.

[62] Reddy, T., Thomas, A. D., Stamatis, D., Bertsch, J., Isbandi, M., Jansson, J., Mallajosyula, J., Pagani, I., Lobos, E. A., and Kyrpides, N. C. The genomes online database (gold) v. 5: a metadata management system based on a four level (meta) genome project classification. *Nucleic acids research* (2014), gku950.

[63] Remolina, S. C., Chang, P. L., Leips, J., Nuzhdin, S. V., and Hughes, K. A. Genomic basis of aging and life-history evolution in drosophila melanogaster. *Evolution 66*, 11 (2012), 3390–3403.

[64] Rivas, M. A., Beaudoin, M., Gardet, A., Stevens, C., Sharma, Y., Zhang, C. K., Boucher, G., Ripke, S., Ellinghaus, D., Burtt, N., et al. Deep resequencing of gwas loci identifies independent rare variants associated with inflammatory bowel disease. *Nature genetics 43*, 11 (2011), 1066–1073.

[65] Rumble, S. M., Lacroute, P., Dalca, A. V., Fiume, M., Sidow, A., and Brudno, M. Shrimp: accurate mapping of short color-space reads. *PLoS Comput Biol 5*, 5 (2009), e1000386.

[66] S, A. Fastqc: a quality control tool for high throughput sequence data. http://www.bioinformatics.babraham.ac.uk/projects/fastqc. Accessed: 2016-03-07.

[67] Schlötterer, C., Tobler, R., Kofler, R., and Nolte, V. Sequencing pools of individuals [mdash] mining genome-wide polymorphism data without big funding. *Nature Reviews Genetics* (2014).

[68] Smith, T. F., and Waterman, M. S. Identification of common molecular subsequences. *Journal of molecular biology 147*, 1 (1981), 195–197.

[69] Smoot, J. C., Barbian, K. D., Van Gompel, J. J., Smoot, L. M., Sylva, G. L., Sturdevant, D. E., Ricklefs, S. M., Porcella, S. F., Parkins, L. D., Beres, S. B., et al. Genome sequence and comparative microarray analysis of serotype m18 group a streptococcus strains associated with acute rheumatic fever outbreaks. *Proceedings of the National Academy of Sciences 99*, 7 (2002), 4668–4673.

[70] Steer, A. C., Law, I., Matatolu, L., Beall, B. W., and Carapetis, J. R. Global emm type distribution of group a streptococci: systematic review and implications for vaccine development. *The Lancet infectious diseases 9*, 10 (2009), 611–616.

[71] Tan, A., Abecasis, G. R., and Kang, H. M. Unified representation of genetic variants. *Bioinformatics* (2015), btv112.

[72] Vallania, F., Ramos, E., Cresci, S., Mitra, R. D., and Druley, T. E. Detection of rare genomic variants from pooled sequencing using splinter. *J Vis Exp 64* (2012), 3943.

[73] Wang, J., Wang, W., Li, R., Li, Y., Tian, G., Goodman, L., Fan, W., Zhang, J., Li, J., Zhang, J., et al. The diploid genome sequence of an asian individual. *Nature 456*, 7218 (2008), 60–65.

[74] Wei, Z., Wang, W., Hu, P., Lyon, G. J., and Hakonarson, H. Snver: a statistical tool for variant calling in analysis of pooled or individual next-generation sequencing data. *Nucleic acids research 39*, 19 (2011), e132–e132.

[75] Wilm, A., Aw, P. P. K., Bertrand, D., Yeo, G. H. T., Ong, S. H., Wong, C. H., Khor, C. C., Petric, R., Hibberd, M. L., and Nagarajan, N. Lofreq: a sequence-quality aware, ultra-sensitive variant caller for uncovering cell-population heterogeneity from high-throughput sequencing datasets. *Nucleic acids research* (2012), gks918.

[76] Wu, P., Zhang, H., Lin, W., Hao, Y., Ren, L., Zhang, C., Li, N., Wei, H., Jiang, Y., and He, F. Discovery of novel genes and gene isoforms by integrating transcriptomic and proteomic profiling from mouse liver. *Journal of proteome research 13*, 5 (2014), 2409–2419.

[77] Xi, H., Park, J., Ding, G., Lee, Y.-H., and Li, Y. Syspimp: the web-based systematical platform for identifying human disease-related mutated sequences from mass spectrometry. *Nucleic acids research 37*, suppl 1 (2009), D913–D920.

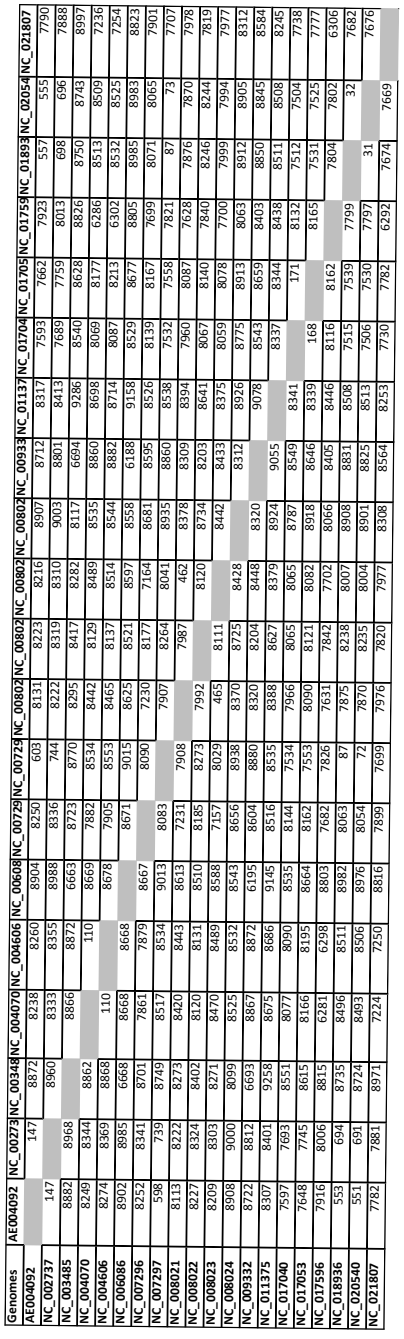
[78] Zhang, Y., Liu, T., Meyer, C. A., Eeckhoute, J., Johnson, D. S., Bernstein, B. E., Nusbaum, C., Myers, R. M., Brown, M., Li, W., et al. Model-based analysis of chip-seq (macs). *Genome biology 9*, 9 (2008), 1.

A. M-types of the strains in the pools

emm type	Non-invasive	Invasive	Total
emm1.0	5	5	10
emm1.10	3	3	6
emm1.24	1	0	1
emm1.45	1	1	2
emm1.50	1	0	1
emm11.0	2	2	4
emm119.1	2	2	4
emm12.0	5	4	9
emm2.0	2	3	5
emm22.0	1	1	2
emm22.3	1	1	2
emm28.0	4	5	9
emm6.4	1	0	1
emm73.0	2	3	5
emm75.0	3	4	7
emm76.3	1	1	2
emm77.0	2	3	5
emm78.3	2	2	4
emm84.0	1	1	2
emm212 (st75.0)	1	1	2
emm11.1	1	0	1
emm4.0	1	1	2
emm78.0	2	2	4
emm89.0	5	5	10
Total	50	50	100

Figure A1: M-types of the strains of the 100 GAS strains used in this study. The M-types have been matched across the pools as much as possible.

B. 20 by 20 pair-wise variants of the GAS genomes



Genomes	AE004092	NC_002737	NC_003485	NC_004070	NC_004606	NC_006086	NC_007296	NC_007297	NC_008021	NC_008022	NC_008023	NC_008024	NC_009332	NC_011375	NC_017040	NC_017053	NC_017596	NC_018936	NC_020540	NC_021807
AE004092		147	8882	8249	8274	8902	8252	598	8113	8227	8209	8908	8722	8307	7597	7648	7916	553	551	7782
NC_002737	147		8960	8344	8369	8985	8341	739	8222	8324	8303	9000	8812	8401	7693	7745	8006	694	691	7881
NC_003485	8882	8960		8968	8868	6668	8701	8749	8273	8402	8271	8099	6693	9258	8551	8615	8815	8735	8724	8971
NC_004070	8249	8344	8968		110	8668	7861	8517	8420	8120	8470	8525	8867	8675	8077	8166	6281	8496	8493	7224
NC_004606	8274	8369	8868	110		8668	7879	8534	8443	8131	8489	8532	8872	8686	8090	8195	6298	8511	8506	7250
NC_006086	8902	8985	6668	8668	8668		8667	9013	8613	8510	8588	8543	6195	9145	8535	8664	8803	8982	8976	8816
NC_007296	8252	8341	8701	7861	7879	8667		8083	7231	8185	7157	8656	8604	8516	8144	8162	7682	8063	8054	7899
NC_007297	598	739	8749	8517	8534	9013	8083		7908	8273	8029	8938	8880	8535	7534	7553	7826	87	72	7699
NC_008021	8113	8222	8273	8420	8443	8613	7231	7908		7992	465	8370	8320	8388	7966	8090	7631	7875	7870	7976
NC_008022	8227	8324	8402	8120	8131	8510	8185	8273	7992		8111	8725	8204	8627	8065	8121	7842	8238	8235	7820
NC_008023	8209	8303	8271	8470	8489	8588	7157	8029	465	8111		8428	8448	8379	8065	8082	7702	8007	8004	7977
NC_008024	8908	9000	8099	8525	8532	8543	8656	8938	8370	8725	8428		8320	8924	8787	8918	8066	8908	8901	8308
NC_009332	8722	8812	6693	8867	8872	6195	8604	8880	8320	8204	8448	8320		9055	8549	8646	8405	8831	8825	8564
NC_011375	8307	8401	9258	8675	8686	9145	8516	8535	8388	8627	8379	8924	9055		9078	8341	8446	8508	8513	8253
NC_017040	7597	7693	8551	8077	8090	8535	8144	7534	7966	8065	8065	8787	8549	9078		8339	8116	7515	7506	7730
NC_017053	7648	7745	8615	8166	8195	8664	8162	7553	8090	8121	7702	8066	8405	8446	8339		168	7539	7530	7782
NC_017596	7916	8006	8815	6281	6298	8803	7682	7826	7631	7842	7702	8066	8405	8446	8339	168		8162	7797	6292
NC_018936	553	694	8735	8496	8511	8982	8063	87	7875	8238	8007	8908	8831	8508	8116	7515	8162		31	7674
NC_020540	551	691	8724	8493	8506	8976	8054	72	7870	8235	8004	8901	8825	8513	7506	7530	7797	31		7669
NC_021807	7782	7881	8971	7224	7250	8816	7899	7699	7976	7820	7977	8308	8564	8253	7730	7782	6292	7674	7669	

Figure B1: The pair-wise variation analysis of the 20 complete GAS genomes.C-Sibelia was used to analyze the variation in these complete whole genome GAS sequences.The smallest number of variants is observed between the GAS genomes AE004092 and NC_002737 and that is because these are the same strains and NC_002737 was updated to AE004092 after correcting some errors and AE004092 is the SF370 strain used as a reference in this study.

C. Alignment and variant calling commands used

```
# preprocessing, remove_adapt.py is an in-house script
remove_adapt.py pool1_lane1_Read1.fastq.gz' pool1_lane1_Read2.fastq.gz'
pool1_lane1_Read1_trimmed_paired.fastq.gz \
pool1_lane1_Read1_trimmed_unpaired.fastq.gz \
pool1_lane1_Read2_trimmed _paired.fastq.gz \
pool1_lane1_Read2_trimmed _unpaired.fastq.gz

# bwa mem alignment
bwa mem AE004092.fasta pool1_lane1_Read1_trimmed_paired.fastq.gz \
pool1_lane1_Read2_trimmed _paired.fastq.gz |
samtools sort -o pool1_lane1_sorted.bam -O bam -T temp -

bwa mem AE004092.fasta pool1_lane2_Read1_trimmed_paired.fastq.gz \
 pool1_lane2_Read2_trimmed _paired.fastq.gz |
samtools sort -o pool1_lane2_sorted.bam -O bam -T temp -

# lane merging after alignment, pool1_header contains
# the sam file headers
samtools merge -rh pool1_header.txt - pool1_lane1_sorted.bam \
pool1_lane2_sorted.bam | samtools sort -o pool1_merged_sorted.bam \
 -O bam -T temp -

# index the merged bam
samtools index pool1_merged_sorted.bam

# variant calling
# Freebayes
freebayes -f AE004092.fasta -m 20 -q 13 -p 40 -F 0.02 \
--use-best-n-alleles 3 --pooled-discrete --pooled-continuous \
pool1_merged_sorted.bam >pool1_freebayes.vcf

# SAMtools
```

```
samtools mpileup -u -q 20 -f AE004092.fasta pool1_merged_sorted.bam |
bcftools call -vm >pool1_samtools.vcf

# SNVer, requires a directory not the bam files directly
java -Xmx64g -jar SNVer-0.5.3/SNVerPool.jar -i pool1_bam/ -r \
AE004092.fasta -n 40 -o pool1_snver -bq 13 -t 0.02

# GATK-UnifiedGenotyper
java -Xmx64g -jar GenomeAnalysisTK-3.4-46/GenomeAnalysisTK.jar -T \
RealignerTargetCreator -R AE004092.fasta -I pool1_merged_sorted.bam -o
\ pool1_target.intervals

java -Xmx64g -jar GenomeAnalysisTK-3.4-46/GenomeAnalysisTK.jar -T \
IndelRealigner - R AE004092.fasta -I pool1_merged_sorted.bam \
-targetIntervals pool1_target.intervals -o pool1_indelrealigned.bam

java -Xmx64g -jar GenomeAnalysisTK-3.4-46/GenomeAnalysisTK.jar -T \
UnifiedGenotyper -R AE004092.fasta -I pool1_indelrealigned.bam \
-ploidy 40 -out_mode EMIT_VARIANTS_ONLY -glm BOTH -stand_call_conf \
20 -stand_emit_conf 20 -o pool1_gatk_noBQSR.vcf

java -Xmx64g -jar GenomeAnalysisTk-3.4-46/GenomeAnalysisTK.jar -T \
BaseRecalibrator -R AE004092.fasta -I pool1_indelrealigned.bam \
-knownSites freeb_gatkNoBQSR.vcf -o pool1_recalibration_report.grp

java -Xmx64g -jar GenomeAnalysisTk-3.4-46/GenomeAnalysisTK.jar -T \
PrintReads -R AE004092.fasta -I pool1_indelrealigned.bam -BQSR \
pool1_recalibration_report.grp -o pool1_gatk_BQSR.bam

java -Xmx64g -jar GenomeAnalysisTk-3.4-46/GenomeAnalysisTK.jar -T \
UnifiedGenotyper -R AE004092.fasta -I pool1_gatk_BQSR.bam -ploidy 50 \
-glm BOTH -stand_call_conf 20 -stand_emit_conf 20 -o pool1_gatk_yesBQSR
```

D. Source code for the create_peptide and retrieve_reads functions

```
def create_peptide(pid,var_info):

    '''Return peptides after including variants.

    Key word arguments:

    pid -- protein id
    var_info -- a list of tuples containing variants
                and their positions from a single read

    '''
    # index_genbank_features() function returns the index of features
    # from the genbank file in this case it is returning the index
    # of coding regions (proteins)

    protein_id_cds_index=index_genbank_features(gbk_file,"CDS",\
    "protein_id")

    # using the protein index, then retrieve additional info
    # such as the aminoacid seq of protein

    index=protein_id_cds_index[pid ]
    cds_feature=gbk_file.features[index]
    immu_tran=cds_feature.qualifiers['translation'][0]
    loc=cds_feature.location
    strand=cds_feature.location.strand
    feature_seq=cds_feature.extract(gbk_file.seq)
    pr_name=cds_feature.qualifiers['product'][0]


    # extracts the nucleotide sequence of the protein
```

```
mut_cds_seq=gbk_file[loc.start:loc.end].seq
mut_cds_len=len(mut_cds_seq)

del_range_list=[]
ins_range_list=[]
modified=False

# var_info has structures (var_pos,alt,ref,effect,'snp'),
# (var_pos,alt,ref,effect,'ins') or
# (var_pos,nt_after_del,alt,ref,'del',effect)

for subset in var_info:
        var_pos=subset[0]
        alt_allele=subset[1]
        ref=subset[2]
        effect=subset[5]

        #since python uses 0-based indexing
        zero_based_pos=int(var_pos)-1

        # since var_pos is interms of the whole sequence
        # and not the protein
        pos_relativeto_cds=abs(loc.start - zero_based_pos)

        # to insert or delete appropriate number of nt
        diff=len(alt_allele)-len(ref)

        if effect == 'START_LOST':
            modified = False
            break

        if diff >= 0: #ins and snp

            # check_if_in_del() returns the modified position
            # if there are deletions preceding this var_pos
```

```
        in_del=check_if_in_del(pos_relativeto_cds,\
        del_range_list)

        if in_del=="True":
            modified=False
            break
        elif in_del=="False":
            pos_relativeto_cds = pos_relativeto_cds
        else:
            pos_relativeto_cds = in_del

        # check_if_in_ins() returns the modified position
        # if there are insertions preceding this var_pos
        pos_relativeto_cds=check_if_in_ins(pos_relativeto_cds,\
        ins_range_list)

        if(diff > 0):
            ins_range_list.append(xrange(pos_relativeto_cds +\
            len(ref),pos_relativeto_cds +\
            len(alt_allele)))
        try:
            mut_cds_seq=mut_cds_seq[:pos_relativeto_cds] + \
            alt_allele + mut_cds_seq[pos_relativeto_cds + \
            len(ref):]
        except IndexError:
            modified = False
            break
        else:
            modified = True


    else: #deletion
        in_del=check_if_in_del(pos_relativeto_cds,\
        del_range_list)
```

```
                if in_del=="True":
                    modified=False
                    break
                elif in_del=="False":
                    pos_relativeto_cds = pos_relativeto_cds
                else:
                    pos_relativeto_cds = in_del

                pos_relativeto_cds=check_if_in_ins(pos_relativeto_cds,\
                ins_range_list)

                del_range_list.append(xrange(pos_relativeto_cds + \
                len(alt_allele),pos_relativeto_cds + len(ref)))

                try:
                    mut_cds_seq=mut_cds_seq[:pos_relativeto_cds \
                    + len(alt_allele)] + mut_cds_seq[pos_relativeto_cds
                    + len(ref):]
                except IndexError:
                    modified = False
                    break
                else:

                    modified = True

        if modified:

            if strand == -1:
                cur_pos=len(mut_cds_seq) % 3
                if cur_pos == 0:
                    initial = loc.start -1
                elif cur_pos == 1:
                    initial = loc.start
                else:
                    initial = loc.start + 1
```

```
        # check_stopcodon_index_backward() returns the stop codon
        # position for the -ve strand seq
        start_index,last_index=check_stopcodon_index_backward\
                                (initial)

        # the modified sequence after the inclusion of the
        # variants,which might be longer or shorter than the
        # original seq. The +1 as backward indices are 1-based
        lengthmodified_cds_seq = gas_seq[last_index +1:loc.start]\
         + mut_cds_seq

        lengthmodified_cds_seq=lengthmodified_cds_seq.\
                                reverse_complement()

    else:
        cur_pos=len(mut_cds_seq) % 3
        initial = loc.end - cur_pos

        # check_stopcodon_index_forward() returns the stop codon
        # position for the +ve strand seq
        start_index,last_index=check_stopcodon_index_forward\
                                (initial)

        lengthmodified_cds_seq = mut_cds_seq + \
        gas_seq[loc.end:last_index]

    mut_tran=str(lengthmodified_cds_seq.translate\
                 (table=11, to_stop=True))

    # the modified sequence in silico digested
    peptide_list_group=re.split(r'([K,R](?!P))',mut_tran)
    peptide_list=[peptide_list_group[i]+peptide_list_group[i+1] \
    for i in xrange(0,len(peptide_list_group)-2,2)]
    if peptide_list_group[len(peptide_list_group)-1] !='':
```

```
                peptide_list.append(peptide_list_group \
                [len(peptide_list_group)-1])

            return (peptide_list)

        else:
            return None


def retrieve_reads(pos_list):
    '''Return sequence reads spanning the list of positions.

    Key word arguments:

    pos_list -- list of variant positions

    '''
    start_pos=pos_list[0]
    end_pos=pos_list[-1]

    sam=subprocess.Popen(["samtools","view","pool_1.sorted.bam",\
    'gi|602625715|gb|AE004092.2|:'+str(start_pos)+"-"+str(end_pos)],\
    stdout=subprocess.PIPE)
    awk=subprocess.Popen(["awk", "-v","OFS=\t",'{print $4,$2,$10}'], \
    stdin=sam.stdout,stdout=subprocess.PIPE)
    sam.stdout.close()
    output=awk.communicate()[0]
    return output
```