Norges miljø- og biovitenskapelige universitet
Fakultet for veterinærmedisin og biovitenskap
Institutt for kjemi, bioteknologi og matvitenskap

Masteroppgave 2014
60 stp

Finding Small Genes by Conservation With a Focus on Bacteriocins

Finne små gener ved konservering med fokus på bakteriociner

Kim Erik Grashei

Preface

In 2010 my co-supervisor, Dzung Bao Diep, graded a master thesis titled "Characterization and regulation of a small stress response protein in *Escherichia coli*" by Ida Hauge at the University of Oslo, which sparked an interest in the search for intergenic bacteriocins. This is how my master thesis came to be.

My work was financed by the institute of Chemistry, Biotechnology and Food Science (IKBM) at the Norwegian University of Life Sciences (NMBU), as well as the biostatistics group and The Laboratory of Microbial Gene Technology group (LMG), both part of IKBM. The work was performed at NMBU in the timespan of january 2013 to may 2014.

Firstly, I would like to thank Lars-Gustav Snipen for going above and beyond what was expected, tirelessly critiquing my work right up to the last minute. It is always enjoyable to come knocking at your office door with my seamlessly endless questions and theories, and ending up discussing them for hours on end!

I would also like to thank my co-supervisor, Dzung Bao Diep, for dragging me out of the dryness of *in silico*, and into the wetness of a laboratory for a few weeks. Seeing both worlds have really put things in a new perspective!

My girlfriend, Janne, also deserves some recognition. Thank you for being there when times were most stressful!

Also, I would like to thank Paweł Oskólski and Marianne Slang Jensen for showing me how to do the lab work, it was greatly appreciated!

Abstract

Gene prediction software is often used to predict genes in genomes through automated annotation pipelines. The success of popular gene finders like Glimmer and GeneMark is reasonably good for long genes, but often fails to predict smaller genes with lengths of 150 nucleotides or less. This is due to the statistical uncertainty associated with predicting small genes. Small open reading frames (ORFs) are expected to appear by chance far more often in a complete genome compared to longer ORFs of 1kb or more.

The goal of this project was to investigate if small genes in bacteria can be found by using conservation, focusing on bacteriocin-producing genes. An algorithm was developed to quantify the conservation of each position in a DNA sequence. Alignments produced by BLAST was analysed in the custom built software *orfstat*, which quantified the conservation of each position of all the analysed genomic sequences.

149 intergenic, i.e. unannotated, chromosome- and plasmid sequences from the *Staphylococcus*- and the *Enterococcus* genera were analysed using BLAST and *orfstat*, and 179 ORFs were selected as bacteriocin gene candidates. Of the 179 candidates, 8 were chosen by manual selection to be tested for antibacterial activity on 53 different bacteria in the laboratory.

When *orfstat* precision was tested on four annotated chromosomes, the RNA-coding annotated regions were given much higher average conservations than the unannotated- and the protein-coding annotated regions. The average protein-coding annotated regions were given about the same average

conservation as the unannotated intergenic regions. The laboratory tests for the eight final bacteriocin candidates did not show any significant inhibition of growth for any of the tested bacteria.

Sammendrag

Genprediksjonsprogrammer er ofte brukt til å predikere gener i genomer gjennom automatiserte annoteringsrutiner. Evnen til populære genfinningsverktøy som Glimmer og GeneMark til å predikere lange gener er rimelig god, men de klarer ofte ikke å predikere mindre gener med lengder på mindre enn 150 nukleotider. Dette er på grunn av den statistiske usikkerheten som eksisterer når det skal predikeres små gener. Små åpne leserammer (ORFer) er forventet å inntreffe mye oftere ved tilfeldighet i en helgenomsekvens sammenlignet med lengre gener på 1kb eller mer.

Målet med dette prosjektet var å finne ut om små gener i bakterier kunne bli funnet ved å bruke konservering, med fokus på bakteriocin-produserende gener. En algoritme ble utviklet for å kvantifisere konserveringen av hver posisjon i en DNA-sekvens. Sammenstillinger produsert av BLAST ble analysert av den selvlagde programvaren *orfstat*, som kvantifiserte konservasjonen av hver posisjon i alle analyserte sekvenser.

149 intergeniske, dvs. uannoterte, kromosom- og plasmidsekvenser fra bakterieslektene *Staphylococcus* og *Enterococcus* ble analysert ved bruk av BLAST og *orfstat*, og 179 ORFer ble valgt ut som bakteriosin-genkandidater. Av de 179 kandidatene ble 8 manuelt utvalgt til å bli testet for antibakteriell aktivitet på 53 forskjellige bakterier i laboratoriet.

Ved testing av fire annoterte kromosomer ble de RNA-kodende annoterte områdene gitt mye høyere gjennomsnittlig konservering enn de uannoterte- og de protein-kodende annoterte områdene av *orfstat* programvaren. Den gjen-

nomsnittlige konserveringsverdien for de protein-kodende annoterte områdene var omtrent lik som for de uannoterte intergeniske områdene. Laboratorietestene for de åtte utvalgte bakteriosin-kandidatene viste ingen signifikant vekstinhibering for noen av de testede bakteriene.

Contents

Chapter 1

Introduction

DNA encodes the genetic instructions for all known organisms. DNA changes over time due to random mutations, which can lead to both small and big phenotypical changes. Most often these mutations have no discernible effect on the organism, and are likely to be passed on to offspring. Despite the lack of publications, mutations are assumed to occur more often in intergenic non-coding regions of DNA compared to coding regions. The intergenic non-coding regions contains no transcribable elements. This is often called *junk DNA*. There are several definitions of junk DNA, but in this thesis the term *real junk DNA* or *real junk* will be used to describe DNA, that when changed, will not give any discernible changes to the organism's fitness.

When an organism experiences a mutation in a real junk region it will continue to live on as before, there is no change in it's fitness. In bacteria this mutation will be passed on to it's daughter cells after binary fission. Each daughter cell will have exactly the same DNA (if we assume no mutations during replication of DNA), including the mutation inherited from the mother cell. The mutation can mutate again in one or both of the daughter cells with no discernible effects. Based on this, an assumption can be made:

Assumption 1. *The frequency of mutations in real junk DNA observed in a population of organisms is only dependent on the physical rules that govern*

all mutations. Rules based on organism fitness are dismissed.

On the other hand, if a mutation occurs in a *non*-real junk region of DNA (e.g. coding regions, promoters) it is much more likely that this mutation will have an effect on the organism's fitness.

If Assumption 1 is true, the real junk regions, or rather the *non*-real junk regions, of DNA can be classified using rules of conservation. In other words, finding a *conserved* site in a DNA sequence means that the site is *not* real junk, that is, it is something of importance to the organism.

Different bacteria contains a wide range of genes, both protein-coding and RNA-coding. These genes resides in the coding regions of chromosomes or plasmids. The non-coding DNA can contain other essential regions, e.g. regulatory elements and structural regions.

Most bacteria have a circular chromosome, and may also contain plasmids. Plasmids are small circular elements of DNA which can be transferred horizontally between some bacteria. Plasmid sizes varies, but in *Enterococcus faecium Aus0085* the size ranges from 2189 bp to 130 716 bp[1], and each bacteria can have multiple plasmids.

This master thesis will mainly focus on protein coding genes. In bacteria, protein coding genes are always *open reading frames* (ORFs). An ORF consists of triplets of nucleotides called codons. The first codon is called the start codon, the last codon is called a stop codon, and codons between the start- and stop codons are non-stop codons. All codons in protein coding ORFs codes for amino acids, except the stop codon.

All protein coding genes are (or contains, subject to the choice of gene definition) ORFs, but not all ORFs are protein coding genes. Real genes most often have regulatory sites associated closely to the ORF. There may also be structural regions in both near- and distant DNA which has an impact on the transcription of genes.

1.1 Bacterial gene finding

Finding genes in bacteria is usually regarded as easier than finding genes in eukaryotic genomes because of the lack of exons and introns in prokaryotic DNA. Repetitive regions can cause problems when searching for genes, but this also has less impact in bacteria because of the smaller non-coding regions[2, 3]. Although the preceding points are true, one of the main problems with finding new genes in bacteria is high intra-species variation, which in some cases can limit the effectiveness of comparative search algorithms[4].

While many genes have been found and annotated, the general opinion is that there still are undiscovered microbial genes[5]. As a means of trying to identify which regions of a bacteria's genome are coding regions, gene prediction if often used. Several gene prediction tools can be used, including Glimmer (http://ccb.jhu.edu/software/glimmer/index.shtml), Prodigal (http://prodigal.ornl.gov/) and GeneMark.hmm (http://exon.gatech.edu/). These gene prediction tools all use different rule sets to identify possible genes. Glimmer uses *interpolated context models* (ICMs) [6]. Prodigal uses a dynamic programming approach consisting of different choices made by the application based on ORFs in the input sequence [5]. GeneMark uses the *Viterbi algorithm for variable duration hidden markov models (HMM)* [7]. These methods, to a certain degree, rely on finding ribosomal binding sites (RBSs), base frequency patterns and the lengths of the open reading frames (ORFs). These predictive methods also often assume that genes are non-overlapping, or that the gene overlap is small (60 bp) [5, 6, 7]. If a real gene is classified by the software as not being a gene, the result is called a false negative. If the software classifies a DNA region to be a gene when it's really not, it's called a false positive.

1.2 The importance of short genes, and why they are hard to find

Bacteria are in a constant state of war with each other over nutrition and space. To win this war, bacteria employ different means to get advantages. One strategy is to kill or inhibit the growth of the surrounding bacteria with antibacterial peptides, such as bacteriocins.

Bacteriocins are peptides produced by a strain of bacteria that are toxic to other strains and species of bacteria[8]. Bacteriocins kill or inhibit the growth of similar or distant bacteria and are usually small peptides with lengths of less than 100 amino acids (aa's), and sometimes less than 30 aa's[9].

The mean protein length for bacterial protein-coding genes is shorter than in eukaryotes [10]. Gene prediction tools are shown to be fairly good at predicting genes with long lengths, with reported correct prediction rates in the range of 70-95% [11]. Since most annotated genes are relatively long with a mean of about 1Kb in bacteria[10], it means that gene prediction tools are generally successful when predicting genes.

However, performing gene prediction on short genes is more tricky. This is because of the statistical uncertainty of classifying a short region of DNA as a coding region. Even though a short region of DNA may contain ORFs, these ORFs are not necessarily coding for anything, and may exist only due to random mutations. See chapter 2.2.2 for general information about ORFs.

Assume that the nucleotides in a DNA sequence are completely random. What is the probability of observing a random ORF with length n?

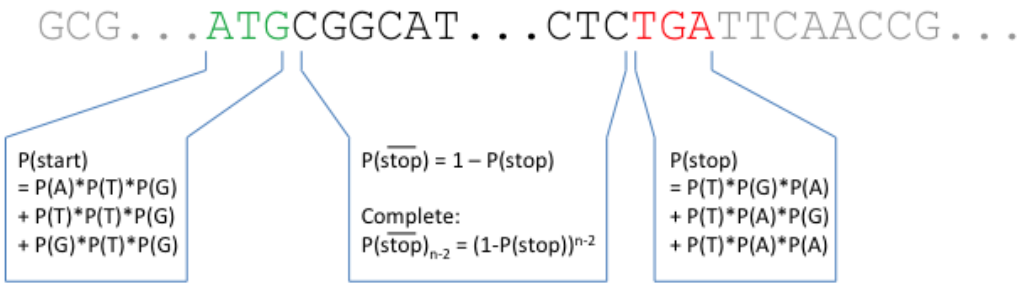

Figure 1.1: Example of an ORF in a random DNA sequence. The start codon is highlighted in green, in this case it is ATG. The box associated with the start codon shows how the probability of observing a random start codon is calculated. The stop codon is highlighted in red showing TGA. The box associated with the stop codon shows how the probability of observing a random stop codon is calculated. The sequence body lies between the start- and stop codons. The associated box shows both how to calculate a single codon which is not a stop codon, $P(\overline{stop}) = 1 - P(stop)$, and the probability of observing a sequence body with $n-2$ codons, $P(\overline{stop})_{n-2} = (1 - P(stop))^{n-2}$. The grey nucleotides to the left and right of the ORF are not associated with the ORF.

As Figure 1.1 shows, the probability of observing an ORF in a random DNA sequence depends on both the start-, and the stop codons. Once a start codon is observed, the length of the ORF is dependent on the probability of observing a stop codon, $P(stop)$. If n is the length of the ORF, then the stochastic variable X is *geometrically distributed*[1], and the probability of observing an ORF with length n is:

$$P(X = n) = P(start) \cdot (1 - P(stop))^{n-2} \cdot P(stop), \quad \text{for } n = 2, 3, ... \quad (1.1)$$

In equation (1.1) n is the number of codons in the ORF and $n-2$ is the body of the ORF, that is, the start codon and the stop codon subtracted from the length of the ORF.

When computing the probability of observing an ORF with length n, the probabilities of observing A's, T's, G's or C's must be known. The bases

[1]Text books often use the form $P(X = n) = (1 - \theta)^{n-1} \cdot \theta$, but X is still geometrically distributed even though a scalar is introduced, as in (1.1).

in a randomly generated sequence are independently, identically distributed (IID), which means that

$$P(A) = P(T) = P(G) = P(C) = 1/4$$

While this is usable when the GC-content of a DNA sequence is not known, in most cases the sequence itself is known, and from it the GC-content. The GC-content is of great importance because of the nucleotides used in start- and stop codons. The three most widely used stop codons in bacteria are TGA, TAG and TAA[12]. There are 4 A's, 3 T's, 2 G's and no C's in these three codons, that is 7 A+T's and 2 G+C's. This means that low GC-content gives a high chance of observing the three stop codons compared to when the DNA sequence has a high GC-content, consequently this also means that a low GC-content produces shorter ORFs by random, and vice versa. The three most widely used start codons in bacteria are ATG, GTG and TTG according to *The Bacterial, Archaeal and Plant Plastid Code* at NCBI[13]. For start codons there are 1 A, 4 T's, 4 G's and no C's. Since there are 5 A+T's and 4 G+C's in the start codons, the GC-content does not have as big of an impact on the occurrences of start codons as on stop codons. High GC-content will give slightly less occurrences of start codons. Probabilities for observing the bases can be constructed based on the GC-content:

$$\begin{aligned} P(A) &= \frac{1-\phi_{GC}}{2} \\ P(T) &= \frac{1-\phi_{GC}}{2} \\ P(G) &= \frac{\phi_{GC}}{2} \\ P(C) &= \frac{\phi_{GC}}{2} \end{aligned} \tag{1.2}$$

Where the ϕ_{GC} is between 0 and 1, and represents the GC-fraction. A ϕ_{GC} of 0.40 means a GC-content of 40 %. The probabilities in (1.2) are the probabilities of observing each base with a GC-content of ϕ_{GC}.

The next step is to calculate the probability of observing one of the three stop-codons, $P(stop)$. Since the probability of observing a base is now given in (1.2), the probabilities for the stop-codons are easily calculated:

$$\begin{aligned} P(TGA) &= P(T) \cdot P(G) \cdot P(A) \\ P(TAG) &= P(T) \cdot P(A) \cdot P(G) \\ P(TAA) &= P(T) \cdot P(A) \cdot P(A) \end{aligned} \tag{1.3}$$

$$P(stop) = P(TGA) + P(TAG) + P(TAA)$$

Where $P(TGA)$, $P(TAG)$ and $P(TAA)$ are the probabilities of observing the stop codons TGA, TAG and TAA respectively, and $P(stop)$ is the probability of observing one of the stop codons.

All codons in a sequence starting with a start-codon, and ending with a stop-codon, are used to construct the length of the ORF. When using the geometric distribution to determine the probabilities for, and expected number of, different ORF lengths, the start codon is assumed to be the first codon.

The expected number of ORFs given ORF length is computed as follows:

$$E_n = P(X = n) \cdot N_{genome}, \quad \text{for } n = 2, 3, ... \tag{1.4}$$

where E_n is the expected number of ORFs observed with length n, $P(X = n)$ is as described in (1.1), N_{genome} is the genome size in codons.

Plots with GC-contents of 30%, 50% and 70% have been constructed in Figures 1.2 and 1.3.

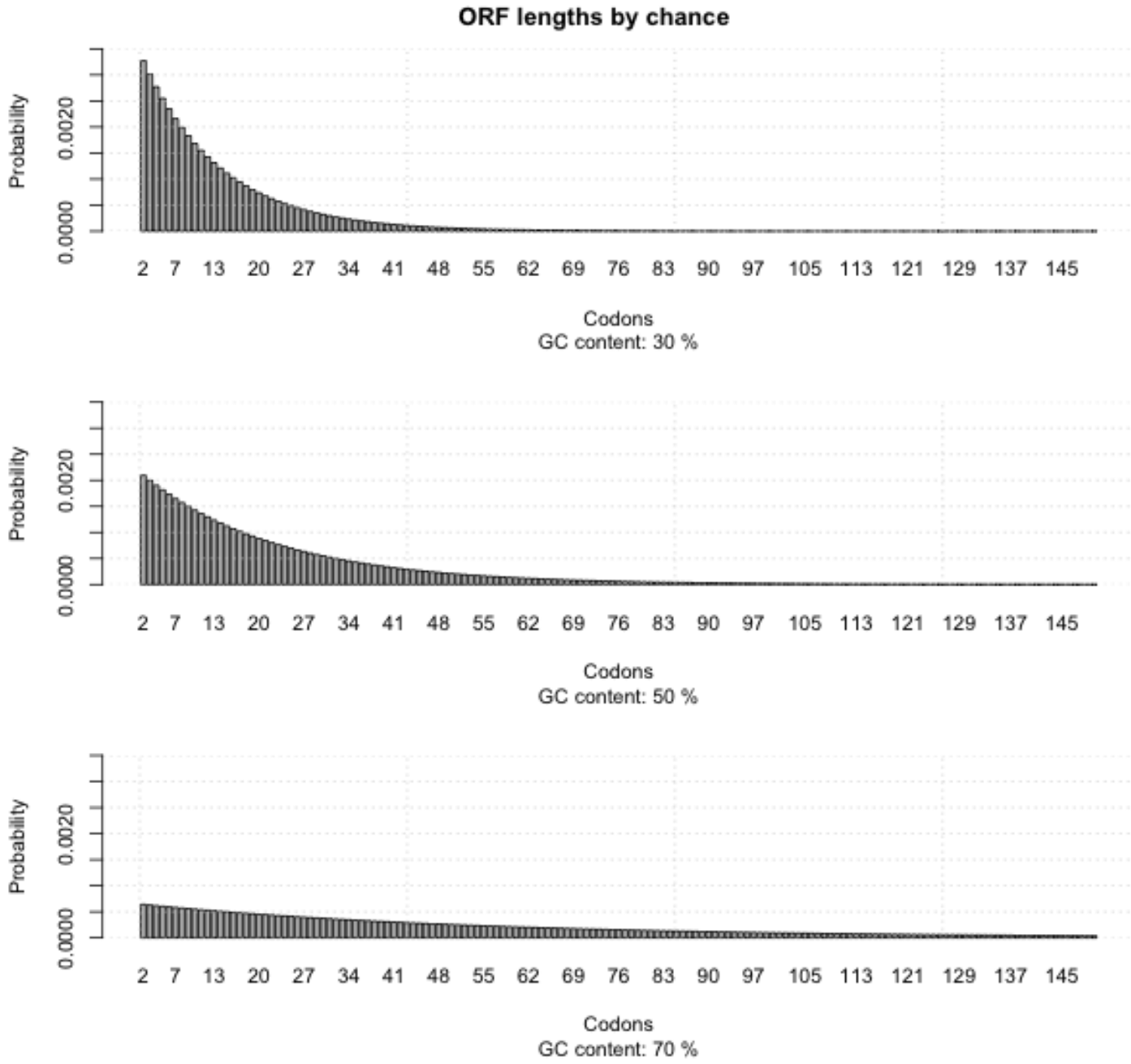

Figure 1.2: Three plots with different GC-contents. The x-axis shows ORF lengths in codons, while the y-axis shows the probability of observing ORFs with the different lengths. Notice that the probabilities of observing longer ORF lengths are higher with a GC-content of 70% compared to a GC-content of 30%.

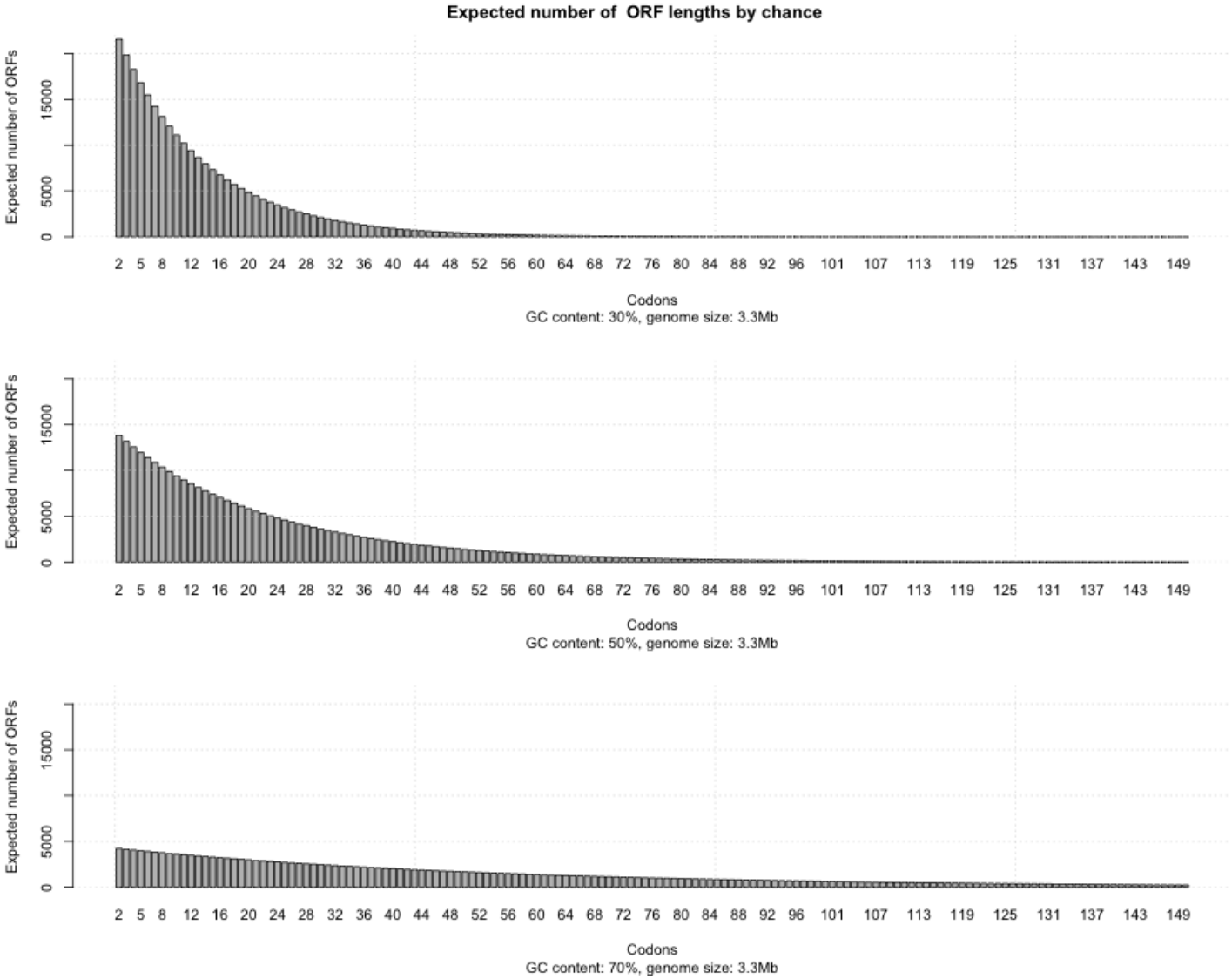

Figure 1.3: The three plots are similar to those in Figure 1.2, but the probabilities are multiplied with a genome length of 3.3Mb, divided by 3 and multiplied with 6, giving the expected number of ORFs given length for a genome size of 3.3Mb. Dividing by three because each codon is a nucleotide triplet, and multiplying by six to get the number of codons on both strands, in all six frames, for the sequence. These are E-value plots for the expected number of ORFs. The x-values are still ORF lengths in codons.

The six plots in Figures 1.2 and 1.3 shows how the distributions for the

ORF lengths are affected by GC-content. Higher GC-content will decrease the probability of observing STOP-codons by chance, and the probability of observing longer ORFs will be higher than with low GC-content.

The probabilities of observing ORF lengths of 15, 25, 50 and 100 codons are about 0.0011%, 0.0007%, 0.0002% and 0.000019% respectively with a GC-content of 50%. On their own, these probabilities may seem small, but with a genome size of 3.3Mb, the expected number of ORFs with these lengths are about 7405, 4582, 1380 and 125 respectively. This makes the process of finding small genes challenging.

1.3 Conservation

Conserved regions in a DNA sequence are regions that have little or no change after many generations of DNA replication. Genes, promoters and structural areas are thought to be noticeably conserved compared to real junk DNA. This is because changes in important regions can be detrimental to the organisms fitness. If the bacteria's fitness declines, it means the chance of survival is lessened, and over time the bacteria with the best fitness will outcompete the others.

Especially the tRNA- and rRNA-coding genes are known to be highly conserved. These genes are found in all known organisms, and are crucial for the organism's ability to synthesize proteins. Conservation of these genes can be seen even at the domain level of biological classification[14, 15].

The general idea is that essential protein coding genes, like the genes involved in creating the DNA polymerase complex, are highly conserved. The DNA polymerase complex is hugely important for all bacteria, and indeed all living organisms, and must be conserved and unaltered for the organism to survive. Conversely, there are genes which are more specialized within one bacterial species, or even within one bacterial strain[16]. In fact a study from 2006 reveals that only 19.7 % of the genes for the pan-genome of *Clostrid-*

ium difficile were shared between the tested strains[17]. These non-shared genes are expected to be more conserved than real junk DNA, but less conserved than the essential genes. Growth inhibiting substances like bacteriocins and other antibacterial peptides are often very specialized towards a certain species or strain, and are therefore not found in many, or even any, other types of bacteria[18]. The genes coding for such peptides are thought to be less conserved than essential genes since these genes are not strictly needed for the bacteria to survive, but they give their host bacteria improved fitness in some environments, and are therefore subject to more change over generations than the essential genes.

Conservation is perhaps most easily studied by analysing big quantities of data, finding regions with high and low mutation rates. Regions with low mutation rates are likely to be conserved, as opposed to the high variability given by frequent mutations in less conserved regions. Unannotated ORFs in regions with low mutation rates are therefore more likely to be genes which have not yet been identified by any other means.

1.4 Testing for antibacterial activity

Most of this thesis revolves around creating algorithms that quantifies conservation of the nucleotides in DNA sequences. As an extension, a laboratory part is added to test if conservation can be used to find ORFs coding for antibacterial peptides, such as bacteriocins.

Antibacterial peptides are, usually small, peptides produced by a strain of bacteria which in some way kills or inhibits the growth of closely- or distantly related bacterial strains or species.

Testing for antibacterial activity is done by cultivating and plating bacteria on agar gel, and adding the candidate peptides to different parts of the plate. The growth, or absence of growth, in different plate regions determines if the peptides have antibacterial activity.

Different bacterial species and strains are used to determine if candidate peptides have an effect in a narrow or wide antibacterial spectrum.

It is important to note that while conservation will be the main method used for finding candidate ORFs, multiple other discriminatory tests must be used when looking for ORFs that are likely to code for antibacterial peptides. These tests include looking at what genes are located upstream and downstream of the ORF (i.e. *gene clusters*), the Shine-Dalgarno sequence and the amphiphilic properties of the candidate peptide.

Gene clustering is especially important to include in the discriminatory search because bacteriocin genes are known to be positioned close to transporter- and immunity genes[18]. Candidate ORFs that are somewhat adjacent to genes of this kind are very interesting.

1.5 Project goals

The focus of this master thesis is to find unknown bacterial genes *in silico* by using conservation. The main goals are as follows:

1. Create algorithms that provides a quantitative prediction of conservation for each nucleotide in a DNA sequence.

2. Develop software that uses the above mentioned algorithms to quantify the conservation values of all nucleotides in an input DNA sequence. This software is called *orfstat* (as in *ORF statistics*).

3. Use orfstat to find ORFs that are candidates for production of bacteriocin peptides.

4. Test if the candidate peptides (bacteriocins) have antibacterial activity in a laboratory.

Chapter 2

Methods

2.1 BLAST

BLAST is a local alignment tool used to align two sequences of nucleotides (nt's) or amino acids (aa's). BLAST is perhaps the most videly applied bioinformatical tool to date, used daily by scientists to find sequence similarities, for species determination and in statistical analyses[19].

BLAST tries to find regions of similarities between two DNA (or peptide) sequences. A local alignment is performed for two sequences at a time, where each alignment is scored by a similarity measure.

In this thesis the BLAST+ software is used to find regions of similarity between an input query sequence, and all subject sequences in a local BLAST database[20]. The output from the BLAST alignments is used to construct a measure of conservation for every position in the input query sequence.

2.2 ORF-finding

A prokaryotic gene is always an ORF, but an ORF is not always a gene.

2.2.1 Obtaining the reading frames

Figure 2.1 shows six full reading frames for a DNA sequence. These six reading frames produce different peptides, all of which can contain zero or more *open* reading frames. All DNA sequences have six reading frames, the first three belonging to the primary DNA strand, and the last three to the complementary strand.

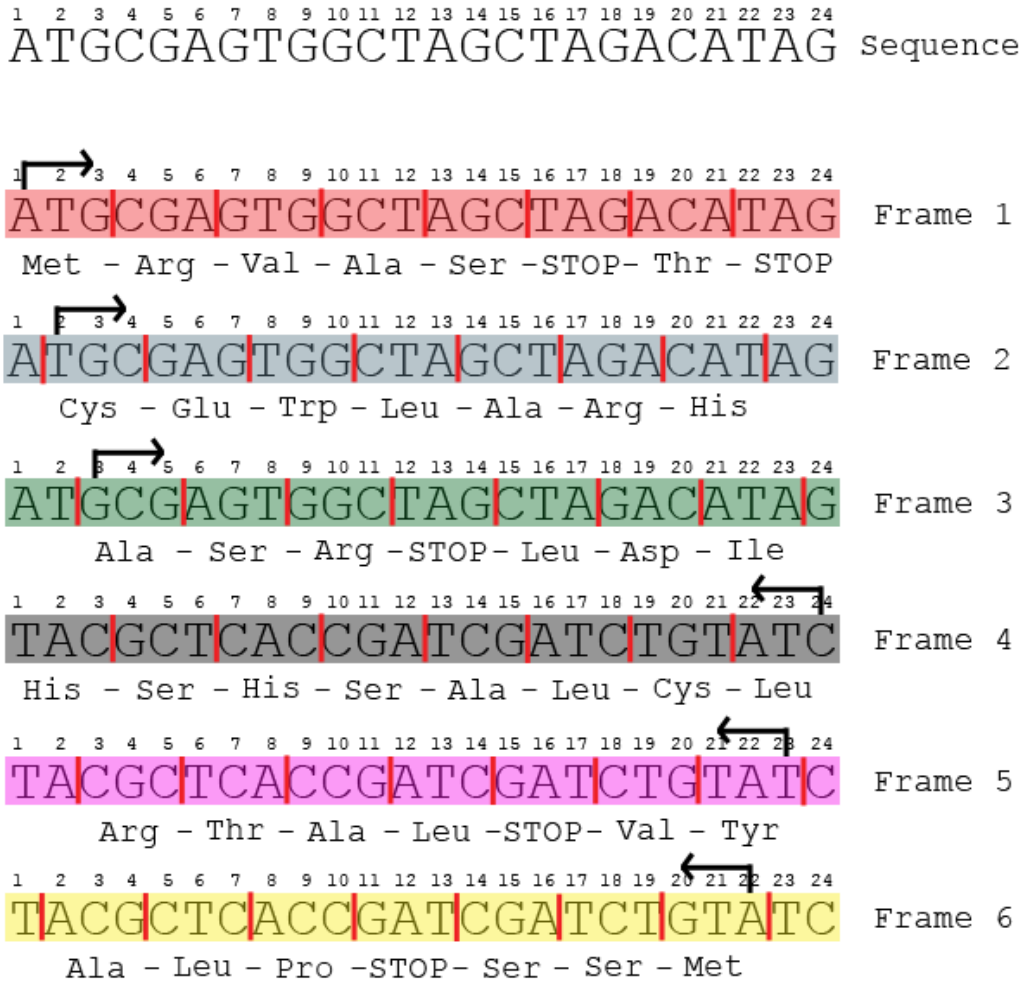

Figure 2.1: Shows all six possible reading frames for a DNA sequence. Vertical red lines indicate codon separations. The small numbers over the sequences indicates nucleotide positions. Black arrows originate from the position of the first codon, and shows the direction of the codon sequence, as well as the read-direction. The corresponding amino acid is indicated below each codon. Frames 1-3 have the same sequence as the original sequence. Frames 4-6 have been made complementary to the original sequence since these frames apply to the complementary DNA strand.

The first reading frame starts at position 1, and ends at position 24. Notice also that there are two ORFs in this frame, one at positions 1-18, and the other at positions 7-18. The first codon starts at position 1, and ends

at position 3. Each codon is a triplet, so the next codon starts at position 4 and ends at position 6. This continues until there are no more codons. The length of the sequence in Figure 2.1 is dividable by three, so it uses all nucleotides in the sequence to construct codons.

The second reading frame starts at position 2 and ends at position 22. The nucleotides at positions 1, 23 and 24 are not used to construct codons, since codons need to be three nucleotides long. Likewise, the third reading frame starts at position 3, and ends at position 23. Positions 1, 2 and 24 are not used.

The fourth to sixth reading frames differ from the first three reading frames. These reading frames are based on the complementary DNA strand, while reading frames 1-3 are based on the primary DNA strand. The strands are therefore made complementary (A's to T's, G's to C's and vice versa). The direction of these sequences is reversed, as indicated by the black arrows in Figure 2.1. Notice that the nucleotide positions remain unchanged. The start position of reading frames 4-6, and the associated peptide sequences, will be larger than the end position. Notice also that in reading frame 6 there is an *open* reading frame from position 22 to position 11.

2.2.2 Finding the *Open* Reading Frames

Open reading frames (ORFs) are important indicators of genes because the coding region of all protein coding prokaryotic genes are ORFs[21]. An ORF starts with a start codon, mostly either ATG, GTG or TTG[13], and ends with a stop codon, mostly either TAG, TGA or TAA[12]. Between the start and stop codons there are codons which codes for different amino acids. A codon codes for a single amino acid. Because codons are triplets of nucleotides, and there are four possible nucleotides at each triplet position, there are $4^3 = 64$ possible codons. The codons code for about 20 different amino acids[13]. Since there are more codons than amino acids, most amino acids are coded by multiple codons, this is called *degeneracy*. Together the

amino acids make up peptides, polypeptides and proteins.

When the reading frames have been determined, and all codons have been translated to amino acids, it's time to find the *open* reading frames. An ORF must start with a start-codon, but may also contain other start-codons which will be part of the ORF. The ORF ends with exactly one stop-codon. If an ORF contains multiple start-codons, multiple ORFs will be constructed, all with their own start codons, but with the same stop-codon.

Start-codons used in this project are ATG, GTG and TTG, and stop-codons are TGA, TAG and TAA.

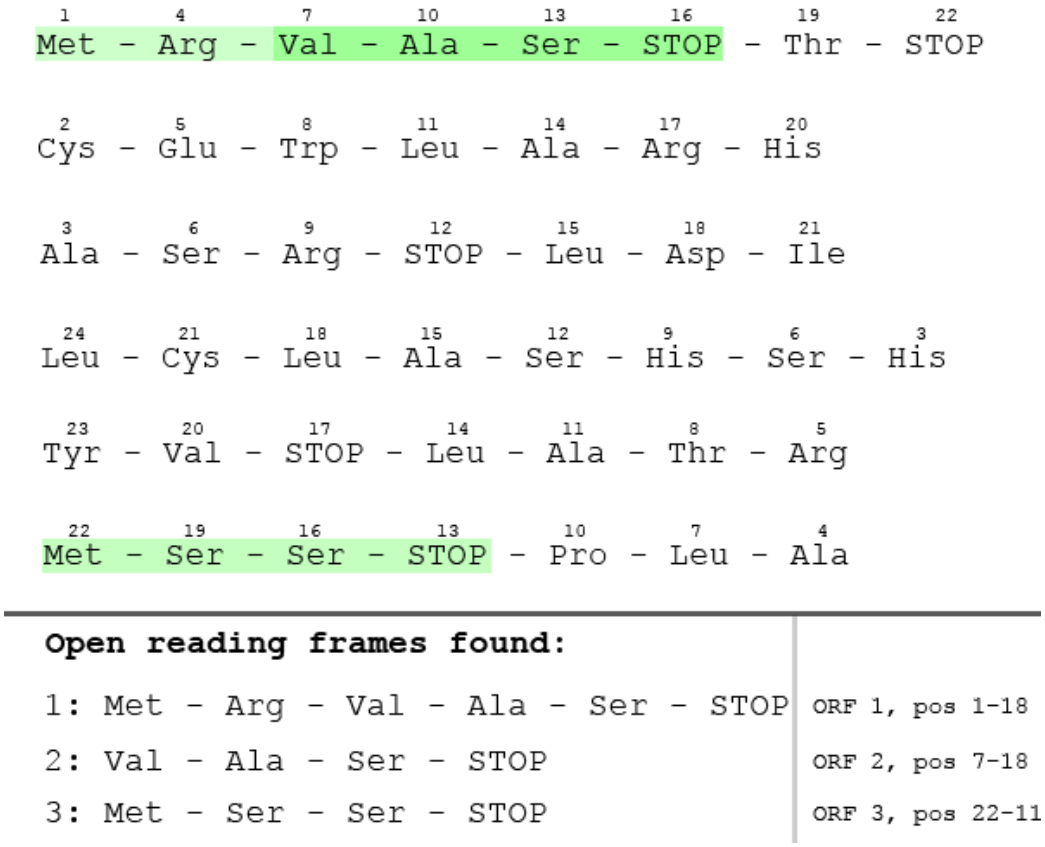

Figure 2.2: Shows the translated peptide sequences for all six reading frames in Figure 2.1. The small numbers above the amino acids are the starting DNA positions for each codon. There are six peptide sequences, corresponding to the six reading frames. There are however only three *open* reading frames, which are found in the first and last reading frames of Figure 2.1. The peptide versions of the ORFs are shown at the bottom of the figure, along with positional information. Note: Even though the second ORF is depicted here as being valine, it is actually methionine when translated in the organism. When translated in an organism, all peptides start with methionine.

Figure 2.2 shows the translated peptide sequences from the DNA-sequences in Figure 2.1. The first amino acid of the first three peptide sequences starts at positions 1, 2 and 3, respectively. If an ORF exists within one of these reading frames, the end position of the ORF must be *incremented* by 2 to include all nucleotides which codes for the peptide sequence. This is shown

at the bottom of Figure 2.2 for the first two ORFs.

Peptide sequences 4-6 originates from the complementary strand, this is why the order of the amino acids is reversed. Notice also that with these peptides, the positions correspond to the primary strand. If an ORF exists within one of these reading frames, the end position must be *subtracted* by 2 to include all nucleotides which codes for the peptide. This is shown for the third ORF at the bottom of Figure 2.2.

2.3 Investigating conservation

Regions containing important DNA, such as genes and regulatory regions, tend to change less in a population of organisms than unimportant "real junk" regions. By studying the individual base similarities, or dissimilarities, between bases in similar regions of DNA, an inference about the conservation of these regions can be made.

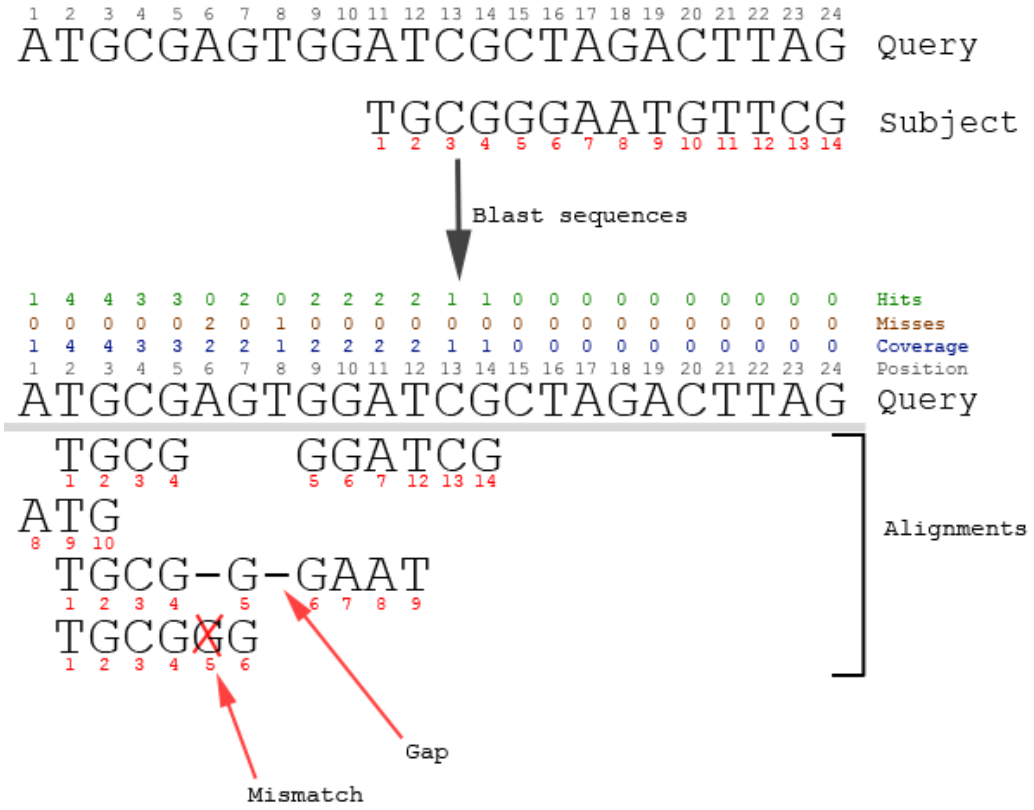

Figure 2.3: A Smith-Waterman alignment algorithm is used in this example to show how a BLAST-alignment might locally align the query DNA sequence (”Query” in the figure) with a subject DNA sequence (”Subject” in the figure). Above the query sequence are grey numbers that indicate nucleotide positions relative to the query sequence. Under the subject sequence there are red numbers indicating nucleotide positions relative to the subject sequence. The vertical black arrow indicates a BLAST alignment of the query- and subject sequences. In this figure the Smith-Waterman aligorithm is used for convenience, with match score of +1, mismatch of -1, and gap penalty of -2. Alignments with score 3 or more was used. ”Hits” shows how many times the alignments have equal bases for a position. ”Misses” shows how many times the alignments have bases which are not equal for a position, this includes both mismatches and gaps in the subject sequence alignment. Coverage is how many times a base in the query sequence has been overlapped by an alignment.

In Figure 2.3 two DNA sequences are aligned locally. The query sequence is always blasted against one, or preferably multiple, subject sequences. The goal is to check for conservation in the query sequence by comparing it to the subject sequences. In the figure, one subject sequence is used to illustrate how conservational information is retrieved (e.g. Hits, Misses and Coverage). In practice, the query sequence is blasted against thousands of subject sequences to produce enough conservational data to find real conserved regions in the query sequence, in this case the hits, misses and coverage of the query sequence will have much higher values. Both the query sequence and the subject sequences may be whole genome sequences, but this is not a requirement.

A few definitions are in order to better understand the coming concepts.

- In the context of coverage information, a base at a position in the query sequence is regarded as a
  - *miss* if the aligned subject sequence contains a mismatch or a gap at this position.
  - *hit* if the aligned subject sequence contains the same base at this position.

Point mismatches, gaps and coverage are included in the term *coverage information.*

The values of *Misses* in Figure 2.3 are incremented when the alignment between the query sequence and the subject sequence produces a mismatch or a gap at a position relative to the query sequence. At position 8 in the query sequence the alignment has produced a gap. Since position 8 is only overlapped once the coverage is 1, and the *Misses* value is 1. Position 6 has coverage of 2 because two alignments overlap this position, but one of the alignments has produced a gap at this position, which then produces a *miss*, and the other alignment has a mismatch that produces another *miss*. The positions with the largest coverage are positions 2 and 3 in the query

sequence. The coverage for these positions is 4, and all alignments in these positions match exactly with the main sequence, so *hits* is also 4. Two nucleotides are not enough to be a gene, so looking beyond the most conserved area can be a good idea, even if the coverage drops somewhat. Positions 4 and 5 are ideal candidates to expand from positions 2 and 3. These positions have a coverage of 3, and *hits* are also 3. Using positions 2 through 5 yields 100 % match for all alignments, with almost equal coverage for all bases. Expanding further will not be easy, and there seems to be little conservation beyond the four nucleotides TGCG. Four nucleotides are not enough for a gene, but it might be enough for a regulatory region, for example.

Any piece of important DNA, which is not real junk DNA, can be searched for by this conservation method, e.g. protein coding genes, RNA-genes, regulatory regions or structural regions. Also, this method is ideal to search for new and unknown elements.

Definitions related to Figure 2.4:

- *Coverage* is the number of times a position in the query sequence has been covered by alignments. Each position in the query sequence has a coverage of zero or more.

- *Mismatches* is the number of times a position in the query sequence has an alignment mismatch with a subject sequence at this position. Each position in the query sequence has zero or more mismatches.

- *Insertion mismatches*, *Insertion mutations* or *Insertions* is the number of times a position in the query sequence has an alignment gap in the subject sequence for this position. Each position in the query has zero or more insertions.

- *Deletion mismatches*, *Deletion mutations* or *Deletions* is the number of times a position in the query sequence contains an alignment gap. Each position in the query has zero or more deletions.

Figure 2.4: Example explaining how to find coverage, hits and misses. The top row consists of numbers indicating position relative to the query sequence, which is the sequence beneath. The complementary query sequence is shown in grey. The two first subject sequences are aligned to the query sequence. The next two subject sequences are aligned to the complementary query sequence, and are shown in light grey. Mismatches are indicated by red letters in the subject sequences. Insertion gaps are shown as red bars, and deletion gaps are shown with a red base with a red arrow indicating the deletion between positions. Coverage is shown, as well as number of point mismatches, insertions and deletions relative to query position. Some sums are shown as well, see text for more information. Vertical grey dotted lines are incorporated for the figure's ease-of-use when comparing numbers at the bottom of the figure to information at the top of the figure.

Figure 2.4 shows an example of a query sequence which is aligned against four subject sequences. The four alignments are shown in the figure as regular text where the subjects are aligned with the query sequence, and grey text where the subjects are aligned with the complementary of the query sequence. Mismatches are shown in red. The red hyphens indicate gaps in the subject sequences. The red arrows with small red bases are gaps in the

query sequence. This method of indicating gaps in the query sequence is used because of the need to show coverage information more easily with respect to query positions, and to use less space. The alternative would be to show the individual alignments between the query sequence and each subject sequence. The small red arrow on the first subject is pointing between positions 38 and 39. This indicates that there should be a gap between these positions in the query sequence. Think of it as "pushing" the small red base in between the subject's positions, and then substituting it with a gap. This is true for all such cases in the figure.

Under the alignments in Figure 2.4 the coverage information can be found. Coverage information exists for all positions in the query sequence. Position 4 (the first position) has only been covered by one alignment, therefore the coverage of this base is 1. Position 5 in the query sequence is covered by three alignments, and so has a coverage of 3, and so on. Notice that the alignments with the complementary sequence is also included in the coverage information. Since coverage is the sum of hits + misses, the coverage of position 18 is 3, even though it has no hits.

Point mismatches are found under the coverage in Figure 2.4. These are regular mismatches, but they can be construed as being possible point mutations in the query sequence. For example, if two bacteria of the same strain had the exact same DNA except for one position which was a point mutation in one of the genome sequences of the two bacteria, this would be represented as a mismatch if these genomes were aligned against each other. Small errors done while sequencing are unavoidable, so mismatches can also just be due to sequencing errors. This is something that is hard to control, so this method assumes all sequencing is "perfect", and that the responsibility of interpreting the results lies with the user. Mismatches may also occur when comparing two different regions, ending up with an alignment that really compares two different sequence elements which has a certain degree of similarity. In Figure 2.4 there is one point mismatch at each of the positions

7, 37, 42 and 43. At position 18 there are two point mismatches.

The insertion mismatches are gaps in the subject sequences. These are indicated by the red hyphens, and in this example there are only insertions related to the queries aligned with the complementary query sequence. All references to point mismatches, insertions or deletions are done with the query sequence in mind. It is perhaps more normal to think of hyphens in an alignment result as deletions. Although this is true, it cannot be known if a deletion in a subject sequence truly is a deletion, since it can also be an insertion in the query sequence. Since this method focuses only on the query sequence, the interpretation of deletions in the aligned subject sequences are though of as insertions in the query sequence. Number of insertions relative to the positions in the query sequence can be seen in the row marked "Insertion mismatches" in Figure 2.4. There is one insertion at each of the positions 14, 17, 18, 19, 28, 29 and 32.

Deletions in Figure 2.4 are represented by using red vertical arrows that point between two bases, also indicating which base has been deleted in red. This is the same as introducing a gap in the query sequence between the two adjacent bases (indicated by the red vertical arrow), and also inserting the base (marked in red) at this position in the subject sequence, which then has an insertion. The base-arrow scheme is used for compacting the figure, and only showing the query sequence as a continuos sequence, one single time. There is an inherent fault with looking at deletions in the query sequence. Since the query sequence is the only sequence of interest when using this conservation method, the query positions are very important because possible mutations are linked to the positions in this sequence. What is the position of a deletion? One might think of this as an earlier version of the query sequence, before the deletion occurred. While this thinking is intuitive, it's no good for analysing the sequence in question. The proposed earlier version of the sequence has another positional scheme, and this scheme cannot be easily used with the current version of the sequence. More on this in chapter

2.3.1. In Figure 2.4 there is a row called ”Deletion mismatches”. This row shows number of deletions, or more correct; gaps in the query sequence, for each position in the query sequence. Since there is no real position for gaps in an alignment, the positions of the imaginary earlier version where the base existed is used, but only one position is used even if there are several deletions at once. An example of this is shown at query position 28 in ”Subject 4 (complementary)”, in the figure. There has been a deletion of three bases, ”GTT”, in the query sequence, but the only position where deletions are incremented is position 28, not positions 29 or 30. The reasons for this are discussed in chapter 2.3.1. Since the same position is incremented multiple times it can lead to more deletions at this position than coverage. This is both intuitively and mathematically wrong with reference to the ”Coverage = Hits + Misses” equation, and therefore ”Misses” only includes insertions and point mismatches. *The deletions are therefore not used to find conservation in the query sequence!*

The three last rows of Figure 2.4 shows aggregated information about the possible mutations. The ”Sum mismatches” row shows the sum of point-, insertion-, and deletion mismatches for each position in the query sequence. Notice the sum of position 28, which is 4. This is a higher value than the coverage for this position, which is caused by the three deletions that are added when aligning the query sequence with ”Subject 4 (complementary)”.

The next row is *Points + insertions*. The point mismatches and insertions, for each position in the query sequence, are added together. This is the sum which is used as *Misses* in the equation *Coverage = Hits + Misses*.

The last row shows number of hits per position. Since *Coverage = Hits + Misses*, then *Hits = Coverage - Misses* is also true. This can be checked manually, and this difference is true for all positions with coverage information in the query sequence.

2.3.1 The problem with deletions

As mentioned in chapter 2.3 there are problems when working with deletions with regard to finding conserved areas in a DNA sequence. Deletions have no real positional values, and can be regarded as the insertions in the aligned subject sequence instead.

It is important to stress that the deletions, even though they may be collected and stored, are not used for analytical purposes when using this method of finding conserved areas. Deletions can possibly be used when regarding all subject sequences as possible query sequences, that is performing the whole conservation analysis on a query sequence, then using a subject sequence as the query sequence, putting the original query sequence into the database, and performing the analysis again but on the subject sequence instead of the original query sequence. This analysis could be done on all subject sequences. This thesis will be limited to looking at one query sequence at a time. A quick explanation as to why the deletions are not used is as follows.

1	2	3	4	5	6				7	8	9				10	11	
A	T	G	C	A	T	-	-	-	T	G	C	-	-	-	C	A	Query sequence
A	T	G	C	A	T	G	C	A	T	G	C	A	T	G	C	A	Subject sequence
1	1	1	1	1	1				1	1	1				1	1	Coverage
0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	0	0	Deletions (opt. 1)
0	0	0	0	0	0				1	1	1				1	1	1 Deletions (opt. 2)
0	0	0	0	0	0				3	0	0				3	0	Deletions (opt. 3)

Figure 2.5: The figure shows different possibilities for storing deletions of a query sequence. The first two lines shows the query- and subject sequences respectively. The positions are shown over the query sequence. Coverage shows the coverage of each position in the query sequence. Deletions opt. 1, 2 and 3 shows three different possible ways of storing the deletions relative to the subject's positions.

Normal coverage and positional information is shown in Figure 2.5. The three last rows in this figure shows different ways of looking at deletions rel-

ative to the subject's positions. "Deletions (opt. 1)" is perhaps the most straightforward approach. Here the deletions are shown at the correct positions relative to the subject sequence. The problem with this is that there is no way of connecting the deletions to any positions in the query sequence, since there are no positions where there are deletions. The positions only exist on the subject sequence. This is a major problem for the conservational analysis of the query sequence.

Deletions must be connected to positions in the query sequence. A method of assigning positions to the deletions is to fix them to the neighbouring positions to the right of the deletion area. The figure shows how this is done in "Deletions (opt. 2)". Positions 7, 8 and 9 gets the previous three deletions. As the previous method of storing deletions, this is also not a correct way to go about it. Since it is a fact that the positions 7, 8 and 9 in the query sequence are not deleted, this cannot be the right answer. A previous version of the query sequence, before the deletions occured, could have used this positional scheme, but with the current alignment information there is no way of knowing if these are deletions in the query sequence, or insertions in the subject sequence. There is another deletion area before the end of the sequence, a triplet deletion. If this method of storing deletions is used, there is a need to extend the query sequence until there are no more deletions to be stored, in this case it's one extra space, indicated by the red 1. In this way, a deletion exists without any coverage, which is counter intuitive.

The last row in Figure 2.5 shows a third way of storing deletions. In this method the deletions are all stored on the next available position after the deletion area. All deletions in the deletion area are stored at one position, that is, if there are three deletions after each other, then the next available position in the query sequence will be affiliated with the three deletions. Both deletions at the positions 7 and 10 shows 3 deletions each. This is assumed to be the best way of the three to store deletions. Both the problems of non-existent positions shown in "Deletions (opt. 1)" and the out-of-bounds

problem in "Deletions (opt. 2)" are avoided by doing it this way. This is also how the *orfstat* software stores deletions (see chap. 2.4).

2.4 Predicting mismatches by using Coverage

After collecting coverage information for a query sequence it is possible to construct a statistical model that uses coverage as the explanatory variable and point mismatches or insertions (subject gaps) as the response variable. By examining the data and parameter estimates, predictions can be done to see what regions contain more mismatches than expected, and also what regions contain less mismatches than expected. Regions with less mismatches than what was expected may be conserved.

In order to predict the number of mismatches for a position with known coverage, a model must be fitted to the data. When fitting a statistical model to data, it is important that the model is suited to represent the data in a good way. No model is perfect, so selecting a suitable model should be done with care. The number of mismatches are discrete values, as is coverage, but since they are densely distributed the assumption of a continuous density distribution should be valid.

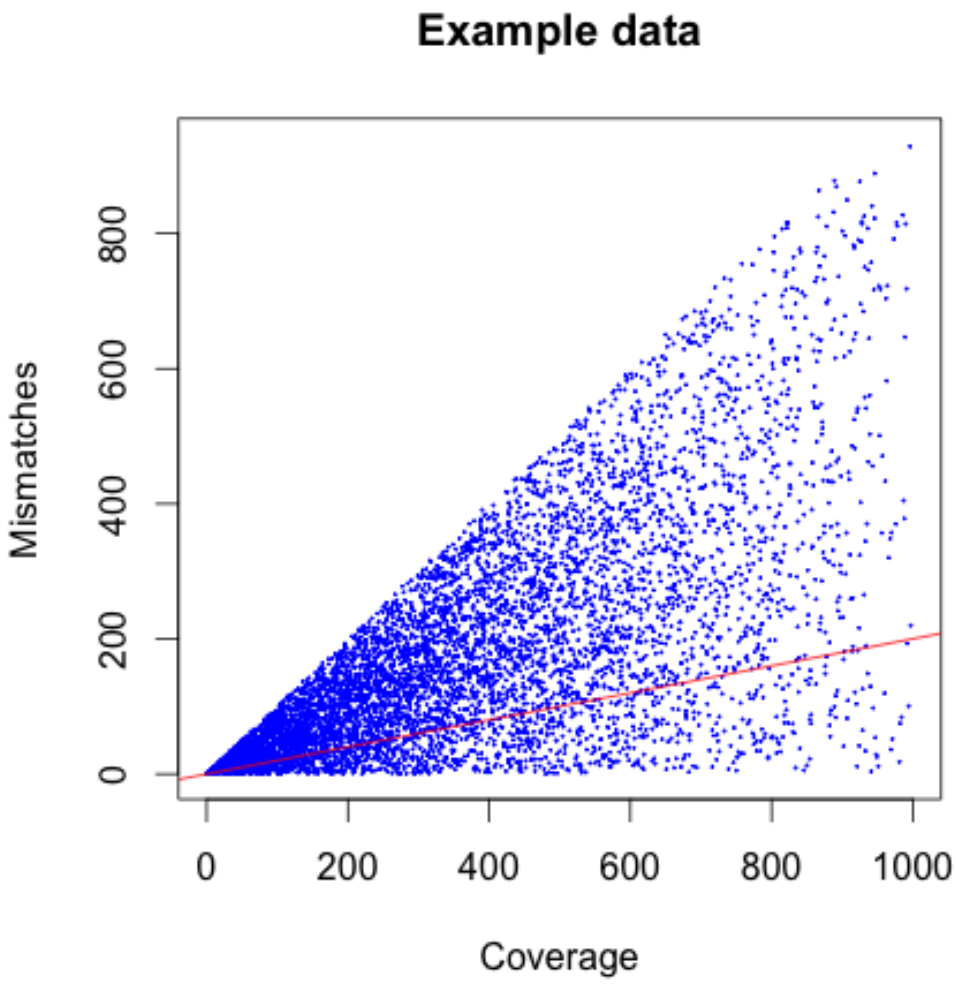

Figure 2.6: The data in the figure is randomly generated. Coverage is shown on the first axis, and alignment mismatches is shown on the second axis. A red regression line has been added to show where the expected number of mismatches can be found for each coverage value. The blue points in the figure are positions in a query sequence. Notice that the number of mismatches cannot be higher than the coverage. Notice also that the variation in mismatches increase as the coverage increases.

As the example data in Figure 2.6 shows, regions of low coverage are inherently worthless since there is not enough data to say anything certain

about these regions, other than that the number of alignments in these regions are scarce. This *could* mean that the low coverage regions are inherently diverse, causing the BLAST search to yield few hits in these regions. This may be interesting to study, but the regions of interest in this thesis are the regions with high coverage and few mismatches.

When the coverage increases, the number of positions will decrease. This is shown in Figure 2.6. A priority should be put on the positions with high coverage, since these contribute more information compared to low-coverage positions. This can be done by using residuals to construct conservation boundaries, as explained next.

The *orfstat* algorithm

The *orfstat* software has been developed solely to be used as an aid for this master thesis. *orfstat* reads the XML-output from BLAST-alignments and calculates the coverage, mismatches, predicted mismatches, mismatch proportion and predicted mismatch proportion for each position in the input sequence, i.e. the query sequence. The mismatch proportion is simply the number of mismatches divided by the coverage for each position. An example output is as follows:

Position	Coverage	Mutations	Pred_mutations	Mut_proportion	Pred_proportion
0	0	0	-48.460367	0.0000000000	0.004351545
1	1027	0	8.052391	0.0000000000	0.007962374
2	1031	0	8.272499	0.0000000000	0.007976438
3	1033	0	8.382553	0.0000000000	0.007983470
4	1033	0	8.382553	0.0000000000	0.007983470
5	1033	0	8.382553	0.0000000000	0.007983470
6	1033	0	8.382553	0.0000000000	0.007983470
7	1033	0	8.382553	0.0000000000	0.007983470
8	1033	0	8.382553	0.0000000000	0.007983470
9	1033	2	8.382553	0.0019361084	0.007983470
10	1033	0	8.382553	0.0000000000	0.007983470
11	1033	0	8.382553	0.0000000000	0.007983470
12	1033	0	8.382553	0.0000000000	0.007983470
13	1033	2	8.382553	0.0019361084	0.007983470
14	1033	0	8.382553	0.0000000000	0.007983470
15	1033	0	8.382553	0.0000000000	0.007983470

Figure 2.7: Output from *orfstat*. *Position* is the position on the intergenic sequence. *Coverage* is how many times the position has been part of an alignment. *Mutations* are how many times each position has had mismatches in alignments. *Pred_mutations* is the predicted number of mismatches for the position. *Mut_proportion* is *Mutations* divided by *Coverage*. *Pred_proportion* is the predicted proportion of mismatches for the position. Both *Pred_mutations* and *Pred_proportion* are predicted using a simple linear regression model, see the text for more information.

As the output in Figure 2.7 shows, both the predicted number of mismatches and the predicted proportion of mismatches for each position is predicted using a *simple linear regression* model:

$$E(y|x) = \beta_0 + \beta_1 x \tag{2.1}$$

where the explanatory variable, x, is coverage and the response variable, y, is either mismatches (*Mutations* in Figure 2.7) or mismatch proportion (*Mut_proportion* in Figure 2.7).

The parameters β_0 and β_1 needs to be estimated. Estimation is done with the *least squares* method, where the goal is to minimize the sum of the

squared residuals, where a residual, e, is defined as:

$$e_i = E(y|x_i) - y_i, \quad \text{for } i = 1, 2, ..., n \tag{2.2}$$

and the optimal β_0 and β_1 are the ones that minimize the sum of the squared residuals, that is:

$$\min \sum_{i=1}^{n} e^2 \tag{2.3}$$

The estimation of β_1 is shown in equation (2.4)[22]:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n}(x_i - \overline{x})^2} = \frac{SS_{xy}}{SS_{xx}} \tag{2.4}$$

Now β_0 can be estimated using $\hat{\beta}_1$:

$$\hat{\beta}_0 = \overline{y} - \hat{\beta}_1 \overline{x} \tag{2.5}$$

Prediction of the number of mismatches and the mismatch proportion is done as follows:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x \tag{2.6}$$

As previously stated, the residual, e, is the number of mismatches that are observed at a given position subtracted from the fitted number of mismatches for the same position. The residuals are used as quantification of conservation for the positions. In this thesis, conservation is thereby defined as follows:

$$Conservation = e = \hat{y} - y \tag{2.7}$$

A positive conservation value for a position indicates that the position is conserved. A negative conservation value indicates that the position is *not* conserved.

There is no absolute boundary for what the conservation value must be for a position to be defined as conserved. This is highly dependent on the

input and BLAST-database when the BLAST-search is performed, as well as the input arguments used when running BLAST. The conservation values are only indications of what *could* be conserved regions.

An open reading frame (ORF) finder has been developed in conjunction with this thesis to find all ORFs in a input sequence (See section 2.2.2). The input sequence in this case is the intergenic sequence. When the ORFs are found, the average conservation of the ORFs are calculated:

$$\text{ORF conservation} = \frac{\sum_{i=j}^{j+(k-1)} e_i}{k} \tag{2.8}$$

where j is the start position of the ORF in the intergenic sequence, k is the length of the ORF, and e_i is the residual (i.e. the conservation) at position i in the intergenic sequence.

Every ORF is given an average conservation number as shown in equation (2.8). The average conservation number is used to sort all ORFs in the input sequence by its average conservation. Sorting is done by the *Collections.sort* method in the Java programming language[23]. When looking at the sorted list, it is assumed to be most valuable to start investigating the most highly conserved ORFs first. The sorting is done automatically by orfstat, and a list of ORFs are returned in a separate file, consisting of both sequence- and conservation data as shown in Figure 2.8.

```
Region: 777420-777397
TCAACCTTGCCGGCGGGGCCCCAA
1710.0500790071535 1619.7336754023722 1618.7598736765833 1381.613461251353 1402.3475190078277 734.3475190078
Sum residuals: 40049.9450920262
Average residuals: 1668.7477121677584

Region: 776271-776266
TTACAA
1840.5395102628572 1837.5395102628572 1483.1499895725417 1082.1499895725417 1816.3447499176993 1758.73427060
Sum residuals: 9818.458020196513
Average residuals: 1636.409670032752

Region: 1437211-1437206
TTACAA
2076.544064969914 2075.544064969914 1813.1545442795987 1286.1545442795987 2057.544064969914 496.836205487650
Sum residuals: 9805.77748895659
Average residuals: 1634.2962481594316

Region: 1952218-1952213
TTACAA
1845.7530559579604 1845.7530559579604 867.168774922487 1658.168774922487 1816.5582956128026 1761.85043613053
Sum residuals: 9795.252393504237
Average residuals: 1632.5420655840396

Region: 2315728-2315723
TTACAA
2064.66368391529 2063.66368391529 1807.274163224975 1276.274163224975 2045.6636839152902 491.9558244330269
Sum residuals: 9749.495202628848
Average residuals: 1624.915867104808
```

Figure 2.8: Output file from orfstat. The file contains information for all ORFs in the analysed sequence. The first line shows the region of the ORF in the input sequence, all ORFs in the figure are found on the complementary DNA strand. The second line shows the sequence itself. The third line lists all the conservation values for the nucleotides in correct order, separated by a whitespace. The sum of all the conservation values in the third line is shown in the fourth line, and the fifth line shows the average conservation value for the ORF, i.e. the sum of the conservation values divided by the length of the ORF (see equation (2.8)). An empty line separates the ORFs.

Figure 2.8 shows a sample of one of the two files produced by orfstat (the other file output is shown in Figure 2.7). This file may be very large depending on the number-, and lengths, of ORFs in the intergenic sequence. Notice that several identical ORFs can be found, and both the primary- and secondary strands are used for ORF finding. Also note that the length of the ORFs can be as short as two codons, these ORFs only contain a start- and a stop codon. Any filtering on the length of the ORFs must be done *a posteriori*.

2.4.1 Step by step description *in silico*

With regard to this thesis, a particular sequence of events have been used repeatedly. The general way of doing things is as follows:

1. Decide on a genome sequence to study, for example a *Staphylococcus aureus* strain. Download the whole genome sequence, as well as the whole genome annotation for the protein-coding genes.

2. Remove the annotated regions from the genome sequence using the genome annotation downloaded in step 1, leaving only the intergenic regions. Make a new sequence out of the intergenic regions, this is the *intergenic sequence*. See Figure 2.9.

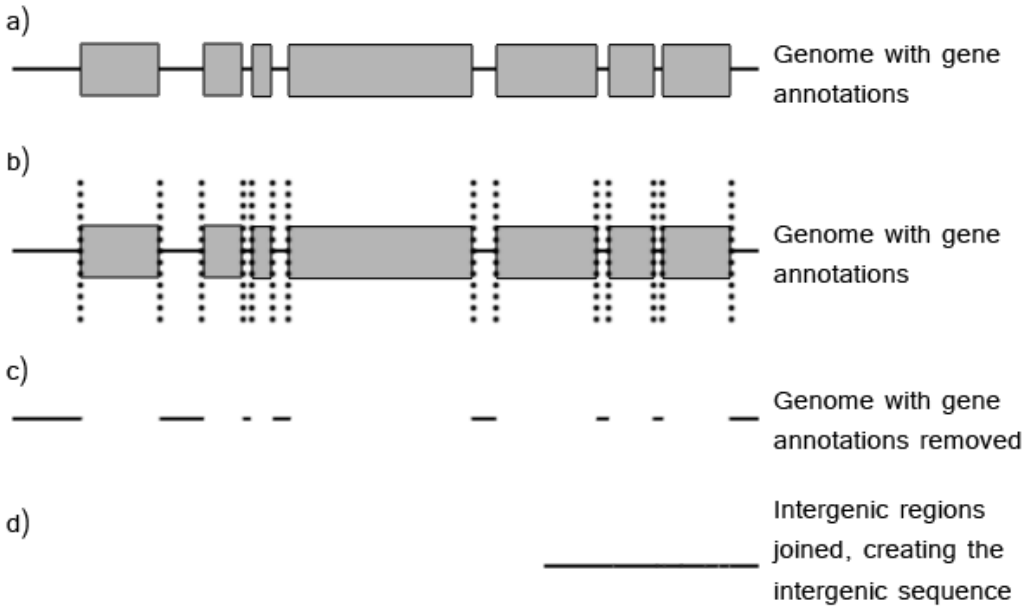

Figure 2.9: Creation of an *intergenic sequence* from an annotated genome. a) Shows the genome as the horizontal black line, and genes are represented as grey boxes. b) Vertical dotted lines are added to show that the annotated regions will be removed. c) Shows the remaining DNA-fragments from the genome sequence. d) The intergenic segments from c) are spliced together and form the *intergenic sequence*.

3. Download all genus-related sequences for the bacterium in question, including whole genome sequences. Make a local BLAST-database of these genus-sequences.

4. Use BLAST to align the intergenic sequence with all sequences from the local genus database. Save the alignment results to an XML-file. *-outfmt 5* is used as a BLAST-parameter to store the alignments in XML-format. An example BLAST-command which performs BLAST-alignments for an intergenic *Enterococcus* sequence against the *Enterococcus* database is as follows:

```
blastn -task megablast -query intergenics_NC_021023.fasta
-db ../../sequences/BLAST_DB/enterococcus_all.fasta
-out blast_results.xml -outfmt 5 -max_target_seqs 10000
-num_threads 1 -dust no -soft_masking false
```

5. Process the alignment results with *orfstat*. Files ending with "_ORFinfo.txt" and "_positionInfo.txt" are created. Information about the ORFs are stored in the "_ORFinfo.txt" file. *orfstat* is called with the default optional arguments.

6. Repeat from step 1 for all species/strains in the study.

7. Filter ORFs from all "_ORFinfo.txt" files with the following conditions *and* order (this is done with separate perl scripts):

   - Remove all ORFs with average conservation less than 50.
   - Translate remaining ORFs to peptides *in silico*.
   - Remove all peptides with sequence lengths less than 15- and more than 50 amino acids.
   - Remove all peptides with *isoelectric point* (pI) less than 9.

- Remove all sequences which are equal or similar.

8. Choose candidate peptides manually from the remaining peptides.

The eight steps is performed on the sequences from the bacterial chromosome- and plasmid sequences listed in Appendix A. Perl scripts were made to automate the process. Perl scripts were also used to perform the filtering in step 7.

2.5 Laboratory part

Laboratory tests are used to find out if any of the chosen candidate peptides really have antibacterial activity.

2.5.1 Materials

The following materials are used to conduct the experiment:

- Agar
- Brain-heart infusion (BHI)
- Distilled water
- 8 candidate bacteriocins, each with concentration 1 mg/mL and purity between 80-95%
- BHT-B bacteriocin, concentration 0.5 mg/mL
- 53 different bacteria (see list below)

Suppliers:

- Peptides were synthesised and supplied by Genscript.

2.5.2 Recipes

Preparation of regular BHI agar, used to make agar-filled plates, is done by mixing 18.5g BHI, 7.5g agar and 500 mL distilled water. This gives half a litre of BHI agar.

BHI soft agar is mixed with bacteria before putting it on top of plates with regular agar. BHI soft agar is made the same way as regular BHI agar, only using half the amount of agar.

BHI growth medium is made the same way as regular agar, except not using any agar.

The three concoctions above all need to be autoclaved before use.

2.5.3 Inhibition assays

Inhibition assays are constructed to see if any of the candidate peptides have antibacterial activity. This is done in the following way:

1. Add regular agar to sterile plates (about 25 mL), let them solidify over night.
2. Streak frozen indicator bacteria on plates to get single colonies. Put in 30 °C over night.
3. Get as many glass tubes as there are plates of bacteria and add 4.5 mL of BHI growth medium to each tube. Take one colony forming unit (CFU) from each plate in the previous step and add it to a tube with growth medium. Put in 30 °C over night. These are clean cultures.
4. Make stock of each clean culture by pipetting 1mL from the glass tubes into a small plastic tube. Add 0.4 mL growth medium and 0.2 mL glycerol. Do this twice, and store at -20 °C and -80 °C, respectively. This is done so it is possible to repeat the experiment with the same bacteria at a later time, if needed.
5. Pipette 100 μL of clean culture into 5mL fluid soft agar. Mix well and add to a clean plate with regular agar. Let it sit 10 minutes to solidify.
6. Pipette 5 μL of each candidate peptide (1 mg/mL) on its own region on the plate. Also pipette 5 μL BHT-B (0.5 mg/mL) to its own region. Let it sit for 10 minutes to dry.
7. Put plates in 30 °C over night.

8. Check if there are inhibitory zones on the plate.

The end result should be something like what is shown in Figure 2.10

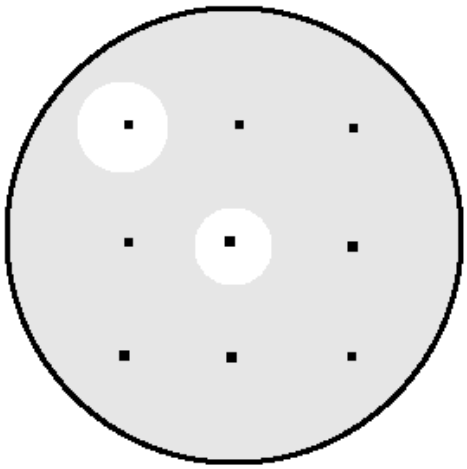

Figure 2.10: Inhibition assay on a plate. The grey background on the plate symbolizes bacterial growth, and the white regions symbolize bacterial growth inhibition. In this example, the middle and top left candidate peptides have inhibited bacterial growth.

The candidate bacteriocins, as well as the BHT-B control bacteriocin, are tested on the following bacteria:

Bacillus cereus LMG 2805
Enterococcus avium LMG 3465
Enterococcus faecalis DEC23 LMGT 3386
Enterococcus faecalis LMG 2333
Enterococcus faecalis LMGT 3358
Enterococcus faecalis SMF37 LMGT 3370
Enterococcus faecium LMG 2722
Enterococcus faecium LMG 2763
Enterococcus faecium LMG 2783
Enterococcus faecium LMG 2876
Escherichia coli LMG 2746[1]
Escherichia coli LMG 3235
L. strain F4-13 LMG 2070
Lactobacillus curvatus LMG 2353
Lactobacillus curvatus LMG 2355
Lactobacillus curvatus LMG 2371
Lactobacillus curvatus LMG 2705
Lactobacillus curvatus LMG 2715
Lactobacillus delbrueckii LMG 3287
Lactobacillus plantarum LMG 2003
Lactobacillus plantarum LMG 2352
Lactobacillus plantarum LMG 2357
Lactobacillus plantarum LMG 2358
Lactobacillus plantarum LMG 2362
Lactobacillus plantarum LMG 2379
Lactobacillus plantarum LMG 3125
Lactobacillus sakei LMG 2334
Lactobacillus sakei LMG 2356
Lactobacillus sakei LMG 2361
Lactobacillus sakei LMG 2380
Lactobacillus sakei LMG 2799
Lactobacillus salivarius LMG 2787
Lactococcus garvieae LMG 3390
Lactococcus lactis IL 1403
Lactococcus lactis LMG 2081
Lactococcus lactis LMG 2130
Lactococcus lactis LMG 3419
Leuconostoc gelidium LMG 2386
Listeria innocua LMG 2710
Listeria innocua LMG 2785
Listeria ivanovil LMG 2813
Listeria monocytogenes LMG 2604
Listeria monocytogenes LMG 2650
Listeria monocytogenes LMG 2651
Listeria monocytogenes LMG 2652
Listeria monocytogenes LMG 2653
Pediococcus pentosacens LMG 2001
Pediococcus pentosacens LMG 2002
Pediococcus pentosacens LMG 2366
Staphylococcus aureus LMG 3022
Staphylococcus aureus LMG 3023
Staphylococcus aureus LMG 3242
Staphylococcus salivarius LMG 1301

[1]This is probably not *E. coli* since the distinct *E. coli* smell was lacking. It also was sensitive to enterocin Q, which *E. coli* should not be due to lack of a target receptor.

Chapter 3

Results

The results presented here are divided into two groups; the main results from *in silico* analyses, and the laboratory results.

3.1 The *in silico* results

The number of possible results produced by *orfstat* are too many to discuss in this thesis. Of the 149 analysed intergenic sequences shown in Appendix A, four are discussed in this thesis, as well as the four whole-chromosome sequences that was used to make the four intergenic sequences, respectively. These four are all shown in Appendix B, and *Enterococcus faecium* Aus0004 is also featured in this results section in figures 3.1-3.7.

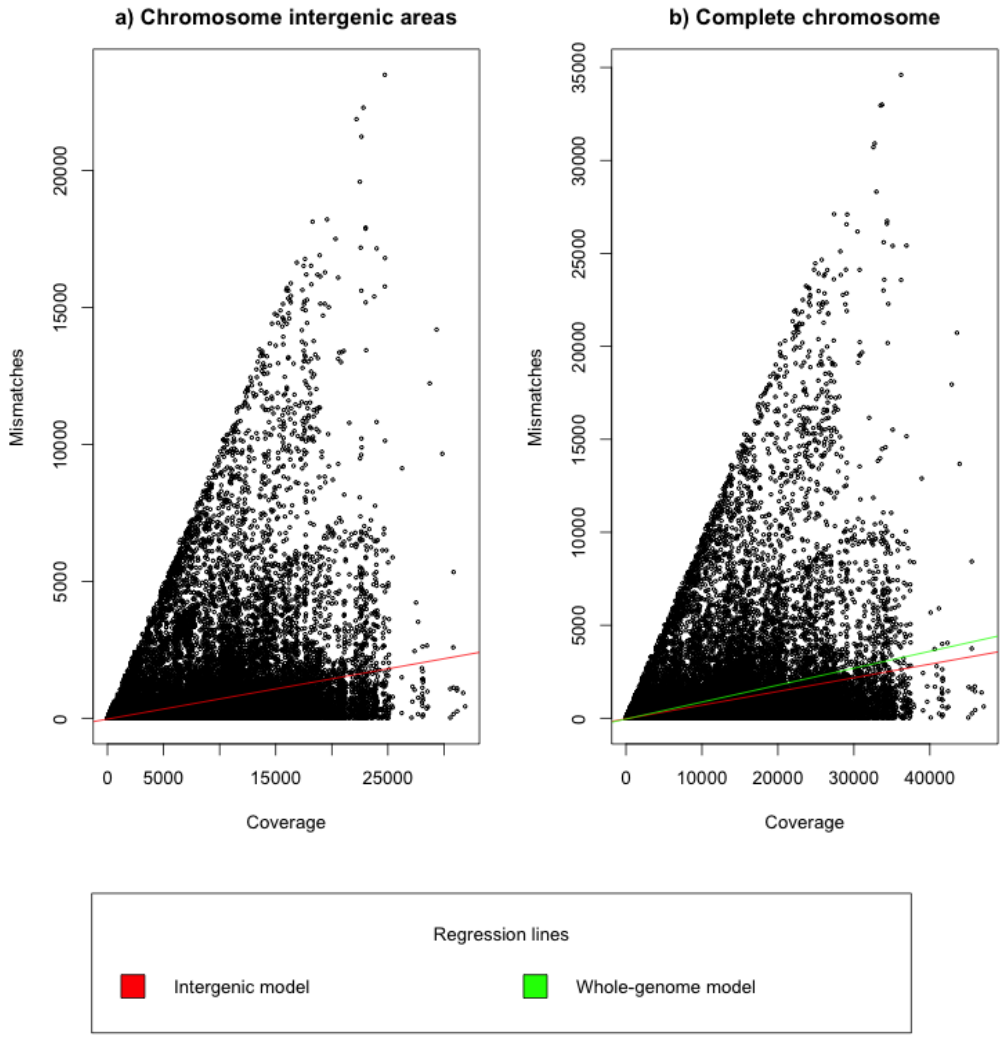

Figure 3.1: The left figure shows the mismatches versus coverage for the intergenic areas of the *Enterococcus faecium Aus0004* chromosome, with a red regression line indicating the expected average number of mismatches. The right figure shows data from the whole *E. faecium Aus0004* chromosome, with a red regression line indicating the expected average number of mismatches based on the intergenic data, and a green regression line indicating the average number of mismatches when the whole chromosome BLAST result is used as data for the model.

The two scatter plots in Figure 3.1 shows the relationship between align-

ment mismatches and coverage in the *E. faecium Aus0004* intergenic sequence (fig. 3.1a), and the whole chromosome (fig. 3.1b). The BLAST alignment results, created from blasting the intergenic- and the whole chromosome sequence from *E. faecium Aus0004* against all available NCBI *Enterococcus* sequences, are processed with *orfstat* to be able to produce Figure 3.1. Figure 3.1a represents the statistical model used in this thesis to quantify the conservation of ORFs in the intergenic *E. faecium Aus0004* sequence. Similar figures *could* be created for all the 149 intergenic sequences studied in this thesis, but only four are shown in Appendix B. Figure 3.1b is used to compare the different regression lines when using intergenic- and whole-chromosome data.

The coverage distributions for the three regions; protein-coding gene annotation, RNA-coding gene annotation and unannotated, are shown in Figure 3.2. The whole-chromosome data is used to construct the figure. Similar results are shown in Appendix B for three other bacteria. Notice that the ranges of the axes for Figure 3.2a) and b) are the same, but differ from c) and d). The coverages for the RNA-coding annotated regions seems to be higher than the other regions.

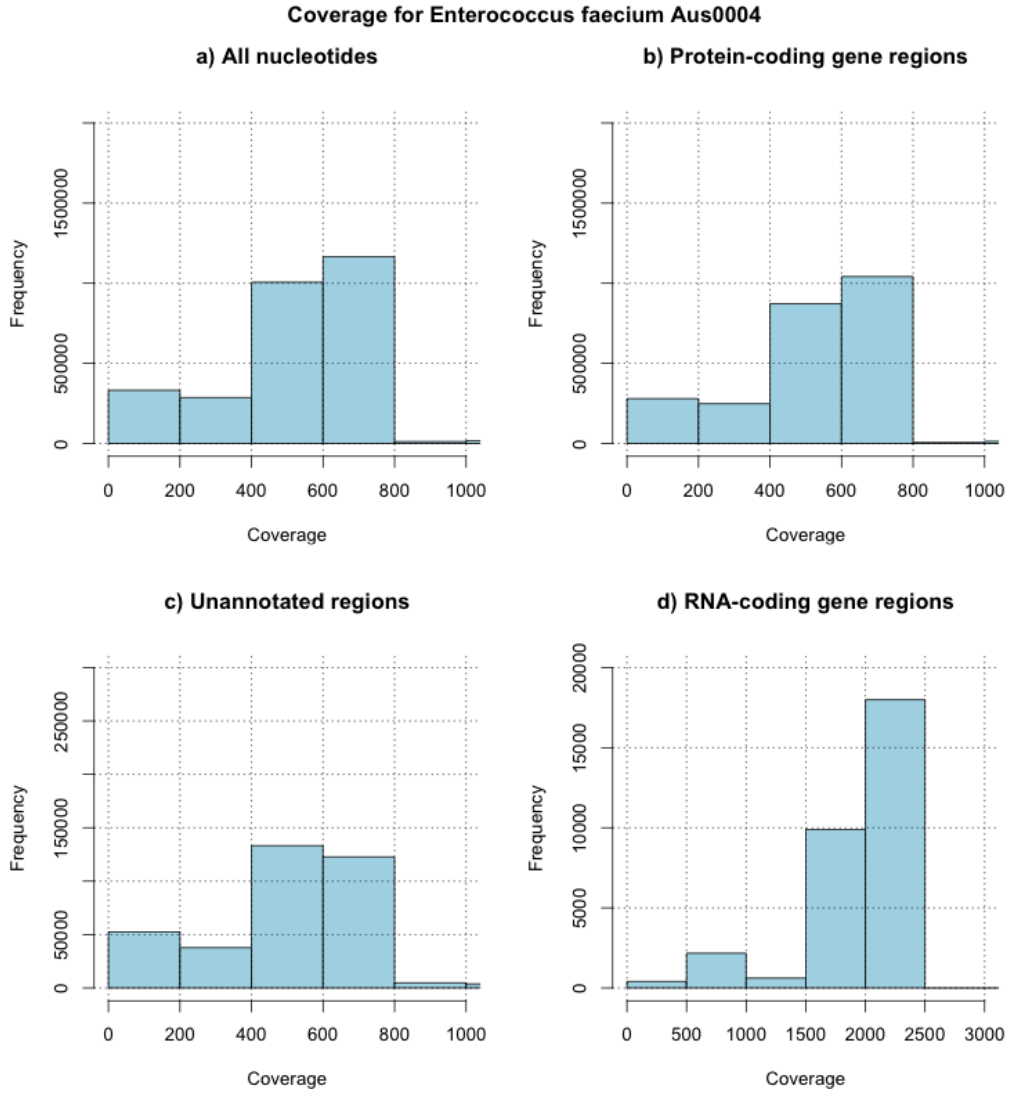

Figure 3.2: Coverage distributions for the nucleic positions in the *Enterococcus faecium Aus0004* chromosome data. a) shows the distribution of coverages for the complete chromosome, while b), c) and d) shows coverage distributions for protein coding- (i.e. the regions with annotations for protein-coding genes), un-annotated- and RNA-coding chromosome regions respectively. Notice that the frequencies shown in a) and b) are higher than in c) and especially d). This stems from the fact that the majority of the chromosome regions are annotated for protein-coding genes.

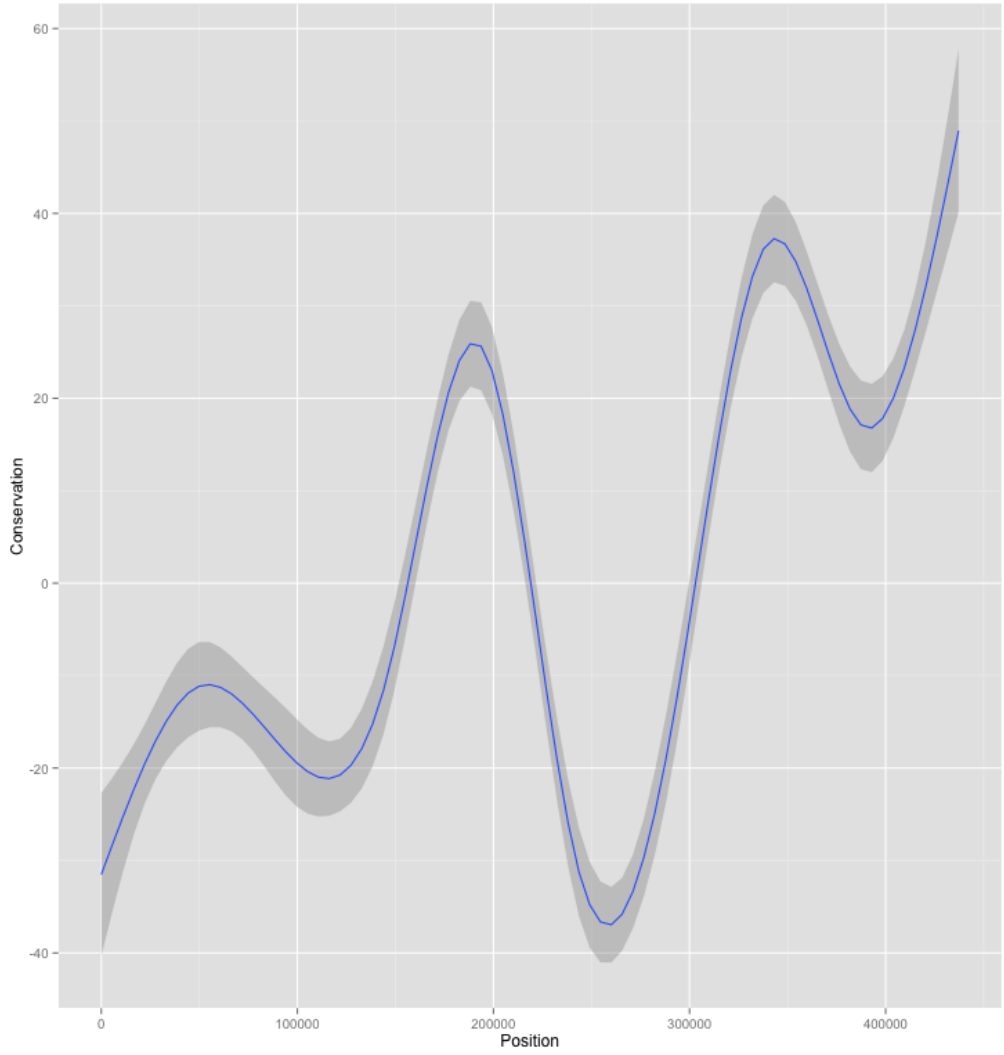

Figure 3.3: Shows the mean conservation by position in the *Enterococcus faecium Aus0004* intergenic data. A 95% confidence interval is shown around the mean line. The figure is generated with the ggplot2-package in R using *stat_smooth* (http://docs.ggplot2.org/0.9.3.1/stat_smooth.html). Default arguments are used, and for datasets with 1000 or more observations like this, the default smoothing model is GAM (http://www.inside-r.org/r-doc/mgcv/gam)

As shown in Figure 3.3, the mean conservation in the *Enterococcus faecium Aus0004* chromosome fluctuates above and below zero. Positions with

a conservation value of zero is neither conserved or un-conserved. Positions with relatively high conservation, e.g. around positions 1,100,000 and 2,400,000 in figure 3.3, are assumed to be more conserved than the rest of the positions in the chromosome. Likewise, the coverages around positions 1,800,000 and 2,900,000 are assumed to be un-conserved.

Figure 3.4 shows a printout from R produced by running the function *TukeyHSD* on the conservation data of the *Enterococcus faecium Aus0004* chromosome. The difference in the mean conservation for the RNA-coding annotations and the protein-coding regions is big, as is the mean differences of the RNA-coding- and the un-annotated regions. The negative sign in Unannotated - RNA-coding is due to the big RNA-coding mean conservation value. The difference, Unannotated - Protein-coding, is not as high as when RNA-coding regions are involved, but still the protein-coding annotations mean conservation is higher than the mean conservation of un-annotated regions. These results are similar to the other analysed sequences in Appendix B.

Notice that the data used to estimate the parameters for the underlying statistical model is intergenic. This means that the intergenic sequence for the *Enterococcus faecium Aus0004* chromosome is blasted against all *Enterococcus* sequences to produce alignments. These alignments are processed by *orfstat*, and the intergenic model parameters are estimated by (2.4) and (2.5). Next, the *Enterococcus faecium Aus0004* whole-chromosome is blasted against all *Enterococcus* sequences. The alignment results are processed by *orfstat* to produce coverages for the whole chromosome. These coverages are use to predict a conservation value for each position in the chromosome, see (2.6) and (2.7).

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = intergenicModel_Conservation ~ Annotations, data = wholegenome_positionInfo)

$Annotations
                                diff        lwr        upr p adj
RNA-coding-Protein-coding    107.30975  103.80359  110.81590     0
Unannotated-Protein-coding   -19.88109  -20.92005  -18.84213     0
Unannotated-RNA-coding      -127.19083 -130.80646 -123.57521     0
```

Figure 3.4: Printout from the *TukeyHSD*-function in R (http://stat.ethz.ch/R-manual/R-patched/library/stats/html/TukeyHSD.html) when used on the *Enterococcus faecium Aus0004* conservation data. Shows the differences between mean conservation value of the groups: Protein coding gene annotation (*Protein-coding* in printout), RNA-coding gene annotation (*RNA-coding* in printout) and unannotated regions (*Unannotated* in printout) under *diff*. Default confidence level is 95%. The statistical model used to predict the conservation values is built on the intergenic data only.

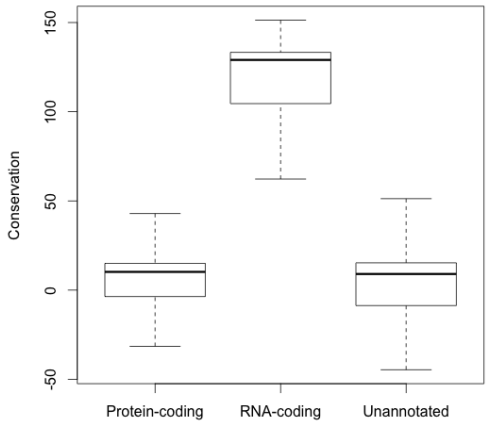

Figure 3.5: Box- and whiskers plot of the conservational values for the nucleic positions in each of the three groups: *DNA annotation*, *RNA annotation* and *Unannotated*. Data is from the orfstat output for the bacterium *Enterococcus faecium Aus0004*. For each group the black bolded line is the median, the horizontal lines under and over the median are the first and third quartiles, repectively. 50 % of the data points are inside the boundaries of this box. R is used to generate the plot, http://stat.ethz.ch/R-manual/R-patched/library/graphics/html/boxplot.html. The statistical model used to predict the conservation values is built on the intergenic data only.

The box- and whiskers plot in Figure 3.5 shows the main spread of conservation values for the three groups when the parameters of the statistical model used for conservation predictions are estimated using intergenic data. The regions which are annotated as RNA-coding has a much higher median than the two other groups. The spread in conservation for the groups seems to be about equal. This result is similar to the other results shown in Appendix B.

The whole *Enterococcus faecium Aus0004* chromosome data was used to build the statistical model used to calculate the results in Figure 3.6, as opposed to only the intergenic data used in Figure 3.4. A smaller difference is observed between the un-annotated regions and the protein-coding annotated regions when using the whole chromosome data. This is consistent with the results shown in Appendix B. The difference is still negative, indicating that the mean conservation for protein-coding regions is larger compared to un-annotated regions. An exception to this can be found for the *Staphylococcus aureus subsp aureus N315* chromosome analysis shown in Appendix B, where the difference between un-annotated- and protein-coding regions is positive, indicating *higher* conservation for the un-annotated regions compared to the protein-coding regions when using the whole chromosome data.

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Conservation ~ Annotations, data = wholegenome_positionInfo)

$Annotations
                                diff         lwr         upr p adj
RNA-coding-Protein-coding    132.131507  128.644261  135.618752     0
Unannotated-Protein-coding    -3.409548   -4.442903   -2.376193     0
Unannotated-RNA-coding      -135.541054 -139.137177 -131.944932     0
```

Figure 3.6: Shows the same information as in Figure 3.4, except the whole chromosome data for *Enterococcus faecium Aus0004* is used to estimate the parameters for the statistical model.

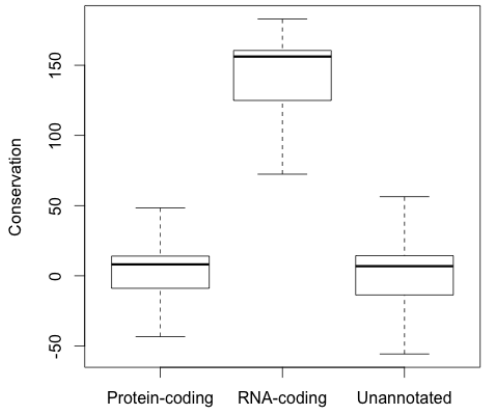

Figure 3.7: Box- and whiskers plot showing the same three groups as in Figure 3.5. The description from Figure 3.5 applies here, except the full chromosome data from *Enterococcus faecium Aus0004* is used to build the statistical model, instead of only the intergenic regions.

The box- and whiskers plot in Figure 3.7 shows the same tendencies as its intergenic counterpart in Figure 3.5, but the RNA-coding regions are slightly less conserved when using the intergenic model compared to the whole-chromosome model. This tendency is also shown in the box- and whiskers plots in Appendix B.

After running *orfstat* on the BLAST results from the intergenic versions of the 149 sequences featured in Appendix A, all ORFs from all sequences were given an average conservation value. This is done by *orfstat* by summing all conservation values in each ORF, and dividing by ORF length. The 1733465 ORFs from all the intergenic sequences needed to be cut down to a manageable amount for manual selection. Table 3.1 shows the filtering steps. Notice that all ORFs are contained in un-annotated intergenic regions *and/or* in RNA-coding regions of the chromosome- and plasmid DNA sequences.

The last step shown in Table 3.1 removes similar peptides from the list. Similarity was found by using the java framework *BioJava* (`http:`

Table 3.1: ORF filtering results. The table shows how many ORFs remain after filtering. The filtering is done stepwise from left to right in succession. Initially, before filtering, there are 300249 ORFs in the intergenic sequences of *Enterococcus* genomes, and likewise 1433216 ORFs in Staphylococcus intergenics. ORF finding and average ORF conservation prediction was done by *orfstat*, and then only keeping the ORFs with an average conservation of 50 or more. All ORFs were then translated to peptides. The remaining filtering was done in the following steps: 1. Keep only peptides that have lengths of 15-50 amino acids. 2. Keep only peptides with theoretical isoelectric point (pI) larger or equal to 9. 3. Remove all peptides that are completely equal. 4. Remove similar peptides.
* One peptide was removed in *Staphylococcus* due to a malfunction in one of the filtering steps (unknown), which resulted in a malformed sequence. This sequence was the second most lowly conserved peptide of all *Staphylococcus* peptides with conservational values equal to or higher than 50.

	ORFs	Conservation ≥ 50	Only 15-50 aa's	pI ≥ 9	Equals removed	Similars removed
Enterococcus	300249	16487	5318	3339	480	84
Staphylococcus	1433216	97763	35356	21665	1238	95*

//biojava.org/wiki/Main_Page), and performing a protein alignment using the BLOSUM 62 matrix. Peptides with alignment similarity of 30% or more was removed, and the peptide with the highest conservation was retained.

Of the 179 final ORFs/peptides shown in Appendix C, only the following eight were chosen to be tested for antibacterial activity in the laboratory:

- Candidate 5:
  - Name: *Staphylococcus aureus subsp. aureus LGA251*
  - Accession: NC_017349
  - DNA container: Chromosome
  - Theoretical pI: 9,26
  - Average ORF conservation: 561,39
  - Translated peptide: MKVYPAQIREWDRNDIFAKFISSSHPNLHIIVS
  - Peptide length: 33 aa's
  - Highest hydrophobic moment: 0,383
  - Upstream 20 nt + start codon: ACTAGCAATAAAGGGTTCAAATG
  - DNA sequence:

ATGAAGGTATATCCAGCTCAAATTAGGGAGTGGGACAGAAATGATA
TTTTCGCAAAATTTATTTCGTCGTCCCACCCCAACTTGCACATTAT
TGTAAGCTGA

- Candidate 10:
  - Name: *Enterococcus faecium NRRL B-2354*
  - Accession: NC_020207
  - DNA container: Chromosome
  - Theoretical pI: 12,28
  - Average ORF conservation: 316,46
  - Translated peptide: LSHALNTYKRRKQKQLLRK
  - Peptide length: 19 aa's
  - Highest hydrophobic moment: 0,288
  - Upstream 20 nt + start codon: GACAAAAGTAAAGAACACTTTTG
  - DNA sequence:

    TTGTCTCACGCTCTAAACACGTATAAACGGCGGAAGCAGAAGCAAC
    TCCTTCGGAAATAA

- Candidate 11:
  - Name: *Enterococcus faecium NRRL B-2354*
  - Accession: NC_020207
  - DNA container: Chromosome
  - Theoretical pI: 9,15
  - Average ORF conservation: 309,50
  - Translated peptide: MFPHIYIFPLIVKNSSSYFAVF
  - Peptide length: 22 aa's
  - Highest hydrophobic moment: 0,304
  - Upstream 20 nt + start codon: CTATTAAGGAAAAATCAACTATG
  - DNA sequence:

    ATGTTTCCCCACATATACATTTTTCCTTTAATTGTGAAAAATAGTT
    CCTCATATTTTGCCGTTTTTTGA

- Candidate 13:
  - Name: *Enterococcus faecium Aus0004*
  - Accession: NC_017022
  - DNA container: Chromosome
  - Theoretical pI: 9,27
  - Average ORF conservation: 250,74
  - Translated peptide: LFGCSYYLMQDSFFTTSFRLALNFLKK
  - Peptide length: 27 aa's
  - Highest hydrophobic moment: 0,385
  - Upstream 20 nt + start codon: AAAATTATGAGGAGCTATTTTTG
  - DNA sequence:

TTGTTTGGCTGTTCTTATTACTTGATGCAGGACAGCTTTTTCACAA
CCTCTTTTCGTTTAGCACTAAATTTCCTTAAAAAGTAG

- Candidate 18:
  - Name: *Staphylococcus aureus subsp. aureus ED133*
  - Accession: NC_017337
  - DNA container: Chromosome
  - Theoretical pI: 9,45
  - Average ORF conservation: 180,99
  - Translated peptide: MYRTTSLTTCASWGGTTK
  - Peptide length: 18 aa's
  - Highest hydrophobic moment: 0,163
  - Upstream 20 nt + start codon: CATACTGATTGAAGACACTAATG
  - DNA sequence:

    ATGTATCGCACCACGTCTCTTACGACATGTGCAAGTTGGGGTGGGA
    CGACGAAATAA

- Candidate 42:
  - Name: *Staphylococcus aureus subsp. aureus LGA251*
  - Accession: NC_017349
  - DNA container: Chromosome
  - Theoretical pI: 10,26
  - Average ORF conservation: 77,00
  - Translated peptide: VWHEVCAISFLLCLRRVSIKKYFFFRN
  - Peptide length: 27 aa's
  - Highest hydrophobic moment: 0,223
  - Upstream 20 nt + start codon: TGAAGCGGTTCAAAAGAAGGGTG
  - DNA sequence:

    GTGTGGCATGAAGTTTGTGCCATATCCTTTTTGTTGTGTTTGCGCA
    GAGTGTCGATAAAGAAATACTTTTTCTTTAGAAATTAG

- Candidate 56:
  - Name: *Staphylococcus aureus subsp. aureus ST228*
  - Accession: NC_020532
  - DNA container: Chromosome
  - Theoretical pI: 10,04
  - Average ORF conservation: 243,74
  - Translated peptide: LRFLCIKKSRKFYLPTIKDEEP
  - Peptide length: 22 aa's
  - Highest hydrophobic moment: 0,276

  - Upstream 20 nt + start codon: ACTTTAAATTATAGAGGCAATTG, ACTTTAAATTATAGAGGCAATTG, ACTTTAAATTATAGAGGCAATTG
  - DNA sequence:

    TTGCGCTTTTTTGTGTATTAAAAAAAGCAGGAAGTTTTACCTTCCCA
    CCATAAAAGATGAAGAACCATAA

- Candidate 57:
  - Name: *Staphylococcus aureus subsp. aureus 6850*
  - Accession: NC_022222
  - DNA container: Chromosome
  - Theoretical pI: 9,30
  - Average ORF conservation: 110,05
  - Translated peptide: MYKNYNMTQLTLPNRNFC
  - Peptide length: 18 aa's
  - Highest hydrophobic moment: 0,073
  - Upstream 20 nt + start codon: CTAAATTAACGAGGTGCCTTATG, CTAAATTAACGAGGTGCCTTATG, CTAAATTAACGAGGTGCCTTATG
  - DNA sequence:

    ATGTATAAAAATTATAACATGACCCAACTTACACTACCCAATAGAA
    ACTTCTGTTAG

Candidates 56 and 57 both have three locations with the exact same DNA sequences in the *Staphylococcus aureus subsp. aureus ST228* and *Staphylococcus aureus subsp. aureus 6850* chromosomes respectively, and thus have three upstream regions each. The upstream regions are also identical for each candidate.

Figure 3.8 shows a screenshot of the candidate 5 neighbouring genes. There is an *ABC transporter* gene upstream from candidate 5. Nearby transporter genes is considered a good indicator of bacteriocin function, since all bacteriocins must be transported out of the cell.

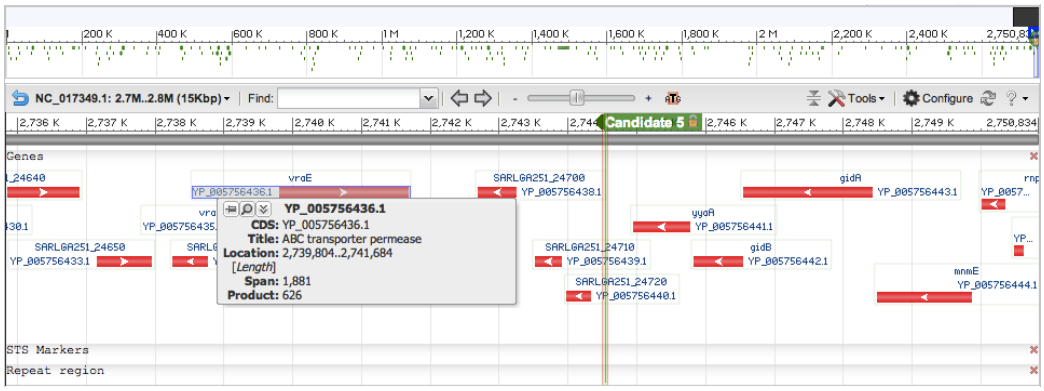

Figure 3.8: Shows candidate 5's neighbouring genes. An upstream ABC transporter is highlighted. Screenshot taken from the NCBI Graphics view at the position of candidate 5 in the *Staphylococcus aureus subsp. aureus LGA251* chromosome sequence.

The hydrophobic moment is a measure of the amphiphilicity of a peptide helix[24]. High hydrophobic moment means that half a turn somewhere in the peptide is hydrophobic, while the other half-turn is hydrophilic. Hydrophobic moments were found using the *heliQuest* helix properties webpage[25]. It takes 18 amino acids to create an analysis window in *heliQuest*, i.e. one α-helical turn of the peptide. The peptide is analysed one 18 aa-window at a time, while shifting the window by one amino acid until the whole peptide is analysed. The part of the peptide with the highest hydrophobic moment is used in the candidate list above.

The last 18 amino acids of the Candidate 13 peptide has a hydrophobic moment of 0.385, which is regarded as moderately high. A helix representation of these amino acids is shown in Figure 3.9.

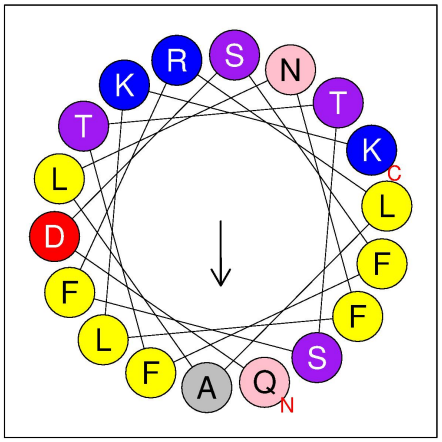

Figure 3.9: The helix generated for the first 18 amino acids in candidate 6. Each circle represents an amino acid, and the letters within the circles are the amino acid one-letter symbols. Yellow amino acids are non-polar, while the other amino acids are polar or special cases. The arrow in the middle of the image is directed towards the hydrophobic region, and its length is decided by the hydrophobic moment. The image is generated by *heliQuest*[25].

The yellow amino acid representations in Figure 3.9 are aligned along the bottom of the helix circle. The bottom half-turn of the helix contains mostly non-polar amino acids except aspartic acid (D), glutamine (Q) and serine (S), and is therefore hydrophobic. The top half-turn contains a grouping of polar amino acids, making it hydrophilic.

Candidate 57 only has 18 amino acids, just enough for one helical turn. The hydrophobic moment is only 0.073, which is regarded as very low. In Figure 3.10, the polar and non-polar amino acids are spread more evenly along the peptide in comparison to Figure 3.9, resulting in a low hydrophobic moment.

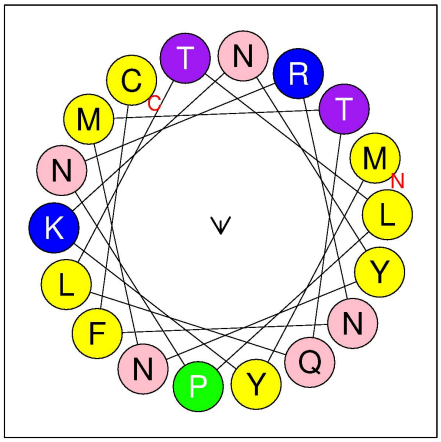

Figure 3.10: The helix generated for the 18 amino acids in candidate 57. The same description applies here as in Figure 3.9. The image is generated by *heliQuest*[25].

High hydrophobic moment was considered advantageous, but was less important than the Shine-Dalgarno sequence when selecting the eight final candidates.

3.2 Laboratory results

The eight candidates were tested for antibacterial inhibition activity using inhibition spectrum assays. The bacteria that were tested are listed in section 2.5.3. No significant inhibition was observed from the candidate antibacterial peptides in any of the assays.

Chapter 4

Discussion

4.1 General

Defining and quantifying conservation for use with bacterial intergenic DNA has been the main focal point of this thesis. Software was developed to test if conservation could be used to find small antibacterial peptide-producing genes in the two genera *Enterococcus* and *Staphylococcus*, the software is called *orfstat*. Two pieces of data are produced by *orfstat*: 1. Positional data for the input sequence, consisting of key conservation data such as coverage and mismatches for each sequence position, and 2. conservation data for each ORF in the analysed input sequences.

orfstat is not strictly gene prediction software. The prediction of conserved areas in DNA sequences can be used to find all genomic elements which are normally conserved, including genes, promoters and RNA- and structural elements. The classification of highly conserved genetic regions must be accompanied by other information relevant to what is searched for, e.g. when looking for genes, ORFs and ORF upstream sequences are highly relevant. Using conservation information is only one of several pieces needed to solve the puzzle.

4.1.1 Using simple regression

As shown in Figure 3.1, *orfstat* produces an enormous amount of data for each analysed sequence. If N is the number of nucleotides in the sequence, there will be $N-2$ degrees of freedom left, where two degrees of freedom are lost to the estimation of β_0 and β_1 in the simple regression model in (2.1).

Using a more advanced method, such as local regression, to more efficiently use the many available degrees of freedom may have proved beneficial. There are two main reasons for only using the simple linear regression model:

1. Time constraints. All statistical models must be implemented in *orfstat*. In addition, each model must be scrutinized and interpreted in the correct way. Using the simple regression model facilitated the development and testing of *orfstat* in a way that made it possible to analyse a great deal of sequences, shown in Appendix A.

2. High-coverage predictions are assumed to be of higher value than predictions done with low coverage. High conservation values cannot occur for positions with low coverages when using the simple linear regression model, but it can occur when using local regression. *orfstat* sorts the list of ORFs by average conservation, so the ORFs with the highest conservation will be first. ORFs with low coverage will be pushed down the list because of the low maximum conservation values they can attain.

4.2 The *in silico* results

4.2.1 Intergenic- vs whole-chromosome analyses

The data shown in Figure 3.1b) is not as expected. It was presumed that annotated regions, both RNA- or protein-coding, would have fewer mismatches compared to the unannotated regions. The red line in Figure 3.1b) shows

the predicted number of mismatches when the parameters of the statistical model (explained in (2.1-2.7)) are estimated using intergenic, i.e. unannotated, data, while the green line uses data from the entire chromosome. If the annotated regions indeed have less mismatches, the green line should have a smaller slope than the red line in the figure. There may be an exponential increase in the number of mismatches when the coverage is very high. Also, outliers have an increasing effect on the data as the coverage increases, which increases the slope of the regression line.

Ideally, the local BLAST databases, one for genus *Enterococcus* and one for genus *Staphylococcus*, should consist of random sequences from these genera. This is probably not the case since the sequences are downloaded from NCBI, and may therefore be biased towards annotated regions. An example of this is shown in Figure 3.2. The RNA-coding regions generally have higher coverage than all other regions, which might have an effect on the results. This is supported by figures 2, 9 and 23 in Appendix B. The variations in the bacterial genomes may also be too diverse to generate hits when using BLAST.

Similar results to the one in Figure 3.1 are found in figures 1, 8, and 22 in Appendix B. All figures display the intergenic regression line beneath the regression line for the whole chromosome. The regression lines are similar for all four bacteria.

4.2.2 Coverage distributions

As mentioned, RNA-annotated regions seems to have higher coverage than the other regions. Figure 3.2d) shows that most of the positions for the *Enterococcus faecium Aus0004* RNA-annotations have a coverage in the interval 1500-2500, as opposed to the unannotated- and protein-coding regions which have coverages around 0-800 shown in Figure 3.2b) and c). All four bacteria shares this trait, as shown in figures 2, 9 and 23 in Appendix B. This indicates that there really is a sequencing bias towards RNA-annotated areas

in the NCBI database. The mean coverages of the *Enterococcus* species are around 600, while it is around 1000 in the *Staphylococcus* species.

4.2.3 Positional chromosome conservation

Figure 3.3 shows that the moving average conservation over the positions in the intergenic sequence of *Enterococcus faecium Aus0004* is not varying by much. The lowest average conservation is about -30, and the highest is around +50. Most of the moving averages of figures 3, 10 and 24 are in the interval between -50 and +50. The lower bound cut off value for the average ORF conservation when filtering was 50 (see Table 3.1), which seems appropriate given the information in the conservation figures. There doesn't seem to be any particular pattern between the average conservation in the plots, except all plots start with a negative conservation. Since there are only four figures to compare, the negative start conservation may only be due to random occurrences.

4.2.4 Mean annotation differences

The mean differences between RNA-annotations, unannotated- and protein-coding annotations for *Enterococcus faecium Aus0004* are shown in the printout in Figure 3.4. The RNA-annotations are on average much more conserved than both the unannotated- and the protein-coding annotations. This is consistent for all four examined bacteria, and shown in the printouts in figures 4, 11 and 25 in Appendix B. Also, the whole chromosome sequence analyses shown in Figure 3.6, as well as figures 6, 13 and 27 in Appendix B, all shows that the RNA-annotated regions are much more conserved than the other regions. Since RNA-coding genes are known to be highly conserved[26, 27, 28], it makes sense that *orfstat* characterizes the RNA-coding annotated regions as being conserved. There is probably a correlation between coverage and conservation because BLAST only shows alignments above a certain score

threshold. This in itself is dependent on how similar the two aligned regions are, so highly conserved regions, e.g. RNA-coding genes, will have more hits because of this than less conserved regions, e.g. some protein-coding genes and unannotated regions.

The difference in mean conservation between protein-coding gene annotations and unannotated regions is substantially smaller than the RNA-coding annotation differences. Figure 3.4 shows that the conservation difference between RNA-coding annotations vs. the unannotated regions is on average about 127 in favor of RNA-coding annotations, and 107 when compared to protein-coding annotations. The difference between protein-coding regions and unannotated regions is only about 20 in favor of the protein-coding regions. A mean conservational difference of 20 may not seem like much, but it still separates the protein-coding regions from the unannotated regions somewhat. The tendencies towards highly conserved RNA-coding annotated regions and low conservation of protein-coding annotated regions for intergenic data are supported by the figures 4, 11 and 25 in Appendix B, admittedly with somewhat lower protein-coding annotation conservations for all examined bacteria, especially for *Staphylococcus aureus subsp. aureus N315* which has a conservation difference of less than 2 for the protein-coding annotated- vs. unannotated regions.

The whole-chromosome Tukey analysis shown in Figure 3.6 shows some conservation difference between the protein-coding annotations and unannotated regions in favor of the protein-coding annotations. The fact that the whole-chromosome Tukey-results shows less conservation for protein-coding annotations than the intergenic results is common for all whole-chromosome analyses shown in figures 6, 13 and 27. In fact, the whole-chromosome Tukey results for *Staphylococcus aureus subsp. aureus N315* have a conservation mean of about 9 in favor of *unannotated* regions. None of the final ORFs in Appendix C comes from either the *S. aureus subsp. aureus N315* intergenic chromosome- or plasmid sequences. In fact, only one ORF from a plasmid

made it into the final ORFs in Appendix C, which may mean that plasmids are generally less conserved than chromosomes.

The boxplots in figures 3.5 and 3.7, as well as figures 5, 7, 12, 14, 26 and 28 in Appendix B, all show that the RNA-coding annotations differ from the unannotated- and the protein-coding annotations in all of the eight analysed sequences (one intergenic- and one whole-chromosome sequence for each of the four bacteria). The boxes pertaining to the unannotated- and protein-coding annotations are similar, visual inspection reveals that protein-coding annotations have a bit smaller variation in conservation than the unannotated regions. Also, there is some visual evidence that the mean conservation is higher in the protein-coding boxes compared to the unannotated boxes, except for the *Staphylococcus aureus subsp. aureus N315* analyses. However, the boxes regarding the unannotated- and the protein-coding annotated regions are very overlapping, and there seems to be little difference between the groups in general. There are three main hypothesised reasons for this:

1. The coverage may be too low in these regions to precisely identify differences in conservation between the groups. Reasons for this may be too few sequences in the BLAST database, or too stringent requirements on the BLAST alignments to get enough alignment hits. Since megablast was used, the latter is probably the main reason for the low coverage seen in unannotated- and protein-coding annotated regions.

2. Annotations are mostly done automatically with prediction of genes and homology searches to provide annotation information[29]. Inaccurate annotations and hypothetical proteins may contribute to the noise level in the protein-coding annotated regions.

3. The *Enterococcus*- and *Staphylococcus* genera may be too diverse to compare different species and subspecies/strains.

4.3 Selection of candidate bacteriocins

The ORF filtering procedure is shown in Table 3.1. All filtering steps are done with perl scripts. The first step simply removes all ORFs with an average conservation lower than 50. Following this, all ORFs are translated *in silico* and treated as candidate peptides. There were more sequences available for *Staphylococcus* species than for *Enterococcus* species, this is the reason that there are many more starting ORFs for *Staphylococcus*.

One of the main goals of this thesis is to investigate if conservation can be used to find small genes in bacteria, with a focus on bacteriocins. This is why only peptides with lengths 15-50 amino acids were retained under the second filtering step.

The isoelectric point (pI) of bacteriocins is often high because the bacteriocins will then segregate towards the negative charge of the phosphate group of the phospholipid-rich cell membranes of the bacteria[30]. The assumption was that a theoretical pI of 9 or more would be enough for peptide segregation towards the cell. Also, it helped drop the total number of candidate bacteriocins to a number that could be worked with manually.

When working with peptides of lengths between 15-50 amino acids, there are bound to be identical peptides. The number of identical peptides was 2859 for *Enterococcus* and 20427 for *Staphylococcus*. Only the peptides with the highest average conservation (determined by the peptide's ORF) were retained when equal peptides were found.

Similar peptides often have the same function. Peptides with an identity similarity of 30% or more was removed, retaining only the peptide with the highest conservation.

The total number of peptides remaining after filtering was 84 for *Enterococcus* and 95 for *Staphylococcus*. The filtering steps were done separately for *Enterococcus* and *Staphylococcus* to ensure that both genera would be represented somewhat equally in the final candidate peptides. All 179 candidate peptides are found in Appendix C. Since only the protein-coding gene

annotations were removed when making the intergenic sequences, not the RNA-coding gene annotations, 101 of the 179 candidates listed in Appendix C overlaps partially or fully with the RNA-coding gene annotated regions in their respective chromosomes. The candidates that overlapped with RNA-coding gene annotations were not used since those regions already have a function that is not known to be antibacterial in nature. There were also 20 candidates which does not exist in the chromosome sequences. This occurs when two intergenic fragments are joined together *in silico* when making the final intergenic sequence from a chromosome. Since these are not really part of the chromosome, they were rejected as possible bacteriocin candidates. This leaves 58 possible candidates for manual selection.

It was considered important that the 20 nucleotide upstream sequence from the gene start codon contained an element with some resemblance to the Shine-Dalgarno motif *GAGG*[31]. It was also considered good if the candidates had the starting codon *ATG*, but this was weighted lower than an optimal Shine-Dalgarno motif. Candidates 5, 11, 18 and 57 have the *ATG* start codon, while candidates 10, 13 and 56 have the *TTG* start codon, and only candidate 42 starts with the *GTG* start codon. All of the selected candidates contain the upstream motif *GAGG* or at most one nucleotide deviating from it, except candidate 10 which has two deviating nucleotides from the *GAGG*-motif; *AAGA*.

Eight peptides were chosen by manual selection to be tested in the lab, these are found in the text within the Results chapter. Candidates 5, 10, 11, 13, 18, 42, 56 and 57 were chosen. They were deemed as plausible bacteriocin candidates.

Another very important criteria is the existence of transporter genes for secretion of the antibacterial peptides. These are often located somewhere in the vicinity of the antibacterial genes[32]. Transporter- or secretory-related proteins exists in the 10Kb upstream- or downstream regions for every of the eight final candidates. Figure 3.8 shows an *ABC transporter* gene upstream

from candidate 5.

Figures 3.9 and 3.10 shows the α-turns of the highest hydrophobic moments of candidates 13 and 57 respectively. High hydrophobic moments suggest that the candidates are amphiphilic, meaning they can permeabilize the bacterial cell membrane and cause lysis of the cell, effectively killing it. Many bacteriocins work in this manner [30]. The hydrophobic property of the candidate peptides were given moderate weight when selecting the final eight candidates.

The reduced priority given to the start codon- and hydrophobic moment properties was because it would impose restrictions on the selection process, making it unlikely to find new types of bacteriocins. The average ORF conservation spanned from 77 (candidate 42) to 561,39 (candidate 5). A wide conservation span was considered prudent since it is hard to say what conservations bacteriocin genes may have in an intergenic sequence analysis, and because this conservation method is new and untested.

4.4 Inhibition spectrum assays

The bacteriocin peptide candidates 5, 10, 11, 13, 18, 42, 56 and 57, as well as the BHT-B control, were tested for growth inhibition properties in the laboratory using inhibition spectrum assays.

The eight bacteriocin peptide candidates and the control were tested on all 53 bacteria listed at the end of the *Methods* chapter. The bacteria spectrum is relatively big, both testing within the *Staphylococcus* and the *Enterococcus* genera, as well as against the *Escherichia coli* and *Leuconostoc gelidium* species, and the *Lactobacillus*, *Lactococcus*, *Listeria* and *Pediococcus* genera. All are gram-positive except for the *E. coli* bacteria.

Enough growth was present on all plates to be able to see inhibition zones, if any. There was no sign of contamination for any of the plates, except for *Escherichia coli LMG 2746* which probably had infections in the

freezed stock bacteria. This was evident because none of the cultures formed from this stock had the distinct *E. coli* smell. An example of how a growth inhibition peptide would stun growth on a plate is shown in Figure 2.10. The BHT-B control showed inhibition zones on most of the plates. BHT-B does not inhibit growth on all bacteria, so this was expected.

No inhibition zones formed for any of the candidate peptides on any of the plates. This indicates that the candidates were not bacteriocins, or at least not bacteriocins that can work independently to inhibit bacterial growth. Most bacteriocins are post-translationally modified before transported out of the cell, making it even harder to find bacteriocins which need no modifications[30, 33].

Bacteriocins are assumed to be less conserved than other genes. This is due to the fact that bacteriocin-producing genes are not as widespread within a population of bacteria as other genes, e.g. genes coding for polymerase subunits. A bacteriocin can in fact be produced by as little as a single bacterial strain[8]. By looking for conserved regions in intergenic sequences, it should be possible to locate bacteriocin genes in the more intermediately conserved regions, depending on the data for producing the alignments. If within-species data is used, instead of within-genus as is done in this thesis, there may be a higher possibility of finding bacteriocin genes.

4.5 Further studies and improvements

Improvements to methods and algorithms are always possible, and this is no exception. Even though there seems to be some merit to searching for conserved regions using BLAST in conjunction with *orfstat*, the results can probably be improved by tweaking the BLAST input parameters, or using another alignment method entirely.

BLAST uses a heuristic search algorithm which decreases sensitivity compared to the Smith-Waterman method[34]. The BLAST algorithm runs up

to 40 times faster than the best known Smith-Waterman implementation[34], this is the main reason BLAST was used in this thesis. Using the Smith-Waterman method could perhaps improve the results produced by *orfstat*. Since *orfstat* imports an XML-file it would be reasonably easy to use other alignment-software than BLAST.

DNA may be too diverse to use with within-genus analyses. Codons are degenerate, meaning multiple codons in an ORF can code for the same amino acid. Combating this issue can be done by aligning proteins instead of DNA sequences. Analysis of protein-alignments has not yet been implemented in *orfstat*.

Because automated annotation pipelines are used when annotating genomes[29], the genomes may, to an unknown degree, be poorly annotated. Removing all hypothetical- and putative protein annotations should be tested to see if the protein-coding gene annotations could be distinguished, in a higher degree than experienced in this thesis, from the unannotated chromosome regions.

Using only one explanatory variable, i.e. coverage, to fit the statistical model used to find conservation of positions in a input sequence may in itself not be enough. Incorporating other variables could increase the conservation prediction accuracy. With multi-variable analyses it would also be possible to see if there are interactions between the variables. Possible variables include:

- Gaps: This thesis only focuses on alignment mismatches. Mismatches can to some extent be construed as being point mutations, since point mutations will cause mismatches. In the same manner, gaps can to a certain degree be interpreted as insertions and deletions. In this way, it could be possible to investigate the interaction effects between point mutations, insertions and deletions.

- Hidden Markov Models (HMM), or models of this nature, can be compared with the conservation data. It would be interesting too see if results from HMMs would correlate with the conservation results.

The use of *local regression* instead of simple regression would be interesting to try, but the conservations calculated by using residuals would not inherently have high coverages for high conservations, as they do with simple regression. This is the major downside to using local regression, or any form of regression where the model is non-linear.

Using within-species data to align sequences could give more precise results when looking for less conserved genes, such as bacteriocin genes. There is much less DNA change within a species compared to within a complete genus, this would produce more alignment hits provided there are enough sequences to align against for only one species of bacteria. Eukaryotes could be more suited for conservation analysis than bacteria since, especially higher, eukaryotic organisms are assumed to have less genetic change over time than bacteria.

Possible uses for conservation analyses done with *orfstat* include:

- Gene discovery. In fact, any genetic elements which are conserved could be found by conservation analysis, including: genes, protein domain families, promoters, structural regions, Shine-Dalgarno motifs etc.

- Evolutionary studies. Distances between species can be predicted by how much they differ in general conservation from the genus, for instance.

- Rate of mutations for different regions. This is essentially what conservation studies are all about; quantifying change in different regions of the genomes. Also, finding out how much more conserved one region is compared to another, e.g. protein-coding regions vs. RNA-coding regions.

- Improve current annotations. A gene annotation has more credibility if it is (very) conserved, especially hypothetical- and putative gene annotations could benefit from this.

4.5.1 Bagel

Bagel[35, 36, 37] is a genome mining tool specifically made to find bacteriocins. The online Bagel tool on `http://bagel.molgenrug.nl/` was unresponsive when trying to analyse intergenic sequences. There were no means of downloading the stand-alone version through the website, and questions asked about it through e-mail was unanswered. Because of this, Bagel has not been used to compare results with *orfstat*.

Bibliography

[1] NCBI. Enterococcus faecium aus0085 plasmids, . URL http://www.ncbi.nlm.nih.gov/nuccore/529235714,529229871,529230034,529230077,529230090,529235635,529205435,529205136,529205379,529205422,529205525,529205299. Accessed: 04/02-2014.

[2] Nicholas Delihas. Impact of small repeat sequences on bacterial genome evolution. *Genome Biol. Evol.*, pages 959–973, 2011. doi: 10.1093/gbe/evr077. URL http://dx.doi.org/10.1093/gbe/evr077.

[3] Guy-Franck Richard, Alix Kerrest, and Bernard Dujon. Comparative genomics and molecular dynamics of dna repeats in eukaryotes. *Microbiol. Mol. Biol. Rev.*, 72:686–727, 2008. doi: 10.1128/MMBR.00011-08. URL http://dx.doi.org/10.1128/MMBR.00011-08.

[4] Lan Ruiting and Peter R. Reeves. Intraspecies variation in bacterial genomes: the need for a species genome concept. *Trends in Microbiology*, 8:396–401, 2000. doi: 10.1016/S0966-842X(00)01791-1. URL http://dx.doi.org/10.1016/S0966-842X(00)01791-1.

[5] Doug Hyatt, Gwo-Liang Chen, and Philip F. LoCascio et al. Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC Bioinformatics*, 2010. doi: 10.1186/1471-2105-11-119. URL http://dx.doi.org/10.1186/1471-2105-11-119.

[6] Arthur L. Delcher, Douglas Harmon, Simon Kasif, Owen White, and Steven L. Salzberg. Improved microbial gene identification with glimmer. *Nucleic Acids Research*, 27(23), 1999.

[7] Alexander V. Lukashin and Mark Borodovsky. Genemark.hmm: new solutions for gene finding. *Nucleic Acids Research*, 26(4), 1998.

[8] B. Lakshminarayanan, C.M. Guinane, P.M. O'Connor, M. Coakley, C. Hill, C. Stanton, P.W. O'Toole, and R.P. Ross. Isolation and characterization of bacteriocin-producing bacteria from the intestinal microbiota of elderly irish subjects. *Journal of Applied Microbiology*, 114: 886–898, 2013.

[9] Yann Héchard and Hans-Georg Sahl. Mode of action of modified and unmodified bacteriocins from gram-positive bacteria. *Biochimie*, 84, 2002.

[10] J. Zhang. Protein-length distributions for the three domains of life. *Trends in Genetics*, 16:107–109, 2000.

[11] Arthur L. Delcher, Kirsten A. Bratke, Edwin C. Powers, and Steven L. Salzberg. Identifying bacterial genes and endosymbiont dna with glimmer. *Bioinformatics*, 23:673–679, 2007.

[12] Inna S. Povolotskaya, Fyodor A. Kondrashov, Alice Ledda, and Peter K Vlasov. Stop codons in bacteria are not selectively equivalent. *Biology Direct*, 7(30), 2012. doi: 10.1186/1745-6150-7-30. URL `http://dx.doi.org/10.1186/1745-6150-7-30`.

[13] NCBI. The bacterial, archaeal and plant plastid code, . URL `http://www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi`. Accessed: 07/02-2014.

[14] Luis E. Gutiérrez-Millán, Alma B. Peregrino-Uriarte, Rogerio Sotelo-Mundo, Francisco Vargas-Albores, and Gloria Yepiz-Plascencia. Sequence and conservation of a rrna and trnaval mitochondrial gene fragment from penaeus californiensis and comparison with penaeus vannamei and penaeus stylirostris. *Mar. Biotechnol.*, pages 392–298, 2002. doi: 10.1007/s10126-002-0008-y. URL `http://dx.doi.org/10.1007/s10126-002-0008-y`.

[15] Jeremy Widman, J. Kirk Harris, and Rob Knight. Stable trna-based phylogenies using only 76 nucleotides. *RNA*, 16, 2010.

[16] Joy Scaria, Lalit Ponnala, and Yung-Fu Chang. Analysis of ultra low genome conservation in clostridium difficile. *PLoS One*, 5, 2010.

[17] R. A. Stabler, D. N. Gerding, J. G. Songer, D. Drudy, J. S. Brazier, H. T. Trinh, A. A. Witney, J. Hinds, and B. W. Wren. Comparative phylogenomics of clostridium difficile reveals clade specificity and microevolution of hypervirulent strains. *J. Bacteriol.*, 188, 2006.

[18] Thomas Lux, Michael Nuhn, and Peter Reichmann. Diversity of bacteriocins and activity spectrum in streptococcus pneumoniae. *J. Bacteriol.*, 189, 2007.

[19] Mark Johnson, Irena Zaretskaya, Yan Raytselis, Yuri Merezhuk, Scott McGinnis, and Thomas L. Madden. Ncbi blast: a better web interface. *Nucleic Acids Research*, 36, 2008. doi: 10.1093/nar/gkn201.

[20] Christiam Camacho, George Coulouris, Vahram Avagyan, Ning Ma, Jason Papadopoulos, Kevin Bealer, and Thomas L. Madden. Blast+: architecture and applications. *BMC Bioinformatics*, 10(421), 2009. doi: 10.1186/1471-2105-10-421.

[21] T. A. Brown. *Genomes 3, 3rd edition*. Garland Science Publishing, 2007.

[22] W. Mendenhall and Terry Sincich. *A Second Couse in Statistics, Regression Analysis, 7th edition*. Pearson Education, Inc., 2011.

[23] Oracle. Java collection class. URL `http://docs.oracle.com/javase/7/docs/api/java/util/Collections.html`. Accessed: 11/04-2014.

[24] David Eisenberg, Robert M. Weiss, and Thomas C. Terwilliger. The helical hydrophobic moment: a measure of the amphiphilicity of a helix. *Nature*, pages 371–374, 1982. doi: 10.1038/299371a0.

[25] heliQuest. Calculation of several helix properties. URL `http://heliquest.ipmc.cnrs.fr/cgi-bin/ComputParamsV2.py`.

[26] C. R. Woese and G. E. Fox. Phylogenetic structure of the prokaryotic domain: the primary kingdoms. *Proc Natl Acad Sci U S A*, 74: 5088–5090, 1977.

[27] C. R. Woese. Bacterial evolution. *Microbiol. Rev.*, 51:227–228, 1987.

[28] Margaret E. Saks and John S. Conery. Anticodon-dependent conservation of bacterial trna gene sequences. *RNA*, 13:651–660, 2007. doi: 10.1261/rna.345907.

[29] Emily J. Richardson and Mich Watson. The automatic annotation of bacterial genomes. *Brief Bioinform.*, 14:1–12, 2013. doi: 10.1093/bib/bbs007.

[30] J. Nissen-Meyer, P. Rogne, C. Oppegard, H. S. Haugen, and P. E. Kristiansen. Structure-function relationships of the non-lanthionine-containing peptide (class ii) bacteriocins produced by gram-positive bacteria. *Current Pharmaceutical Biotechnology*, 10:19–37, 2009. doi: 10.2174/138920109787048661.

[31] Jiong Ma, Allan Campbell, and Samuel Karlin. Correlations between shine-dalgarno sequences and gene features such as predicted expression

levels and operon structures. *J Bacteriol.*, 184:5733–5745, 2002. doi: 10.1128/JB.184.20.5733-5745.2002.

[32] Matthew R. Son, Mikhail Shchepetov, Peter V. Adrian, Shabir A. Madhi, Linda de Gouveia, Anne von Gottberg, Keith P. Klugman, Jeffrey N. Weiser, and Suzanne Dawid. Conserved mutations in the pneumococcal bacteriocin transporter gene, blpa, result in a complex population consisting of producers and cheaters. *mBio*, 2, 2011. doi: 10.1128/mBio.00179-11.

[33] Djamel Drider, Gunnar Fimland, Yann Héchard, Lynn M. McMullen, and Hervé Prévost. The continuing story of class iia bacteriocins. *Microbiol Mol Biol Rev.*, 70:564–582, 2006. doi: 10.1128/MMBR.00016-05.

[34] T. Rognes and E. Seeberg. Six-fold speed-up of smith-waterman sequence database searches using parallel processing on common microprocessors. *Bioinformatics*, 16:699–706, 2000.

[35] Anne de Jong, Sacha A. F. T. van Hijum, and Oscar P. Kuipers. Bagel: a web-based bacteriocin genome mining tool. *Nucleic Acids Research*, 34:W273–W279, 2006. doi: 10.1093/nar/gkl237.

[36] Anne de Jong, Auke J. van Heel, and Oscar P. Kuipers. Bagel2: mining for bacteriocins in genomic data. *Nucleic Acids Research*, 38:W647–W651, 2010. doi: 10.1093/nar/gkq365.

[37] Auke J. van Heel, Anne de Jong, and Oscar P. Kuipers. Bagel3: automated identification of genes encoding bacteriocins and (non-)bactericidal posttranslationally modified peptides. *Nucleic Acids Research*, 41:W448–W453, 2013. doi: 10.1093/nar/gkt391.

Appendix A: Analysed sequences

Table 1: All chromosome- or plasmid sequences analysed with *orfstat*. The sequences are used to find candidate bacteriocin peptides. Note: All sequences are intergenic. None contain any protein-coding gene annotated regions, *but* they do contain RNA-coding annotated regions.

Accession	Genus	Species	Subspecies/strain	Type
NC_020995	Enterococcus	casseliflavus	EC20	Chromosome
NC_018221	Enterococcus	faecalis	D32	Chromosome
NC_018222	Enterococcus	faecalis	D32 EFD32pA	Plasmid
NC_018223	Enterococcus	faecalis	D32 EFD32pB	Plasmid
NC_019770	Enterococcus	faecalis	str. Symbioflor 1	Chromosome
NC_004668	Enterococcus	faecalis	V583	Chromosome
NC_004669	Enterococcus	faecalis	V583 pTEF1	Plasmid
NC_004671	Enterococcus	faecalis	V583 pTEF2	Plasmid
NC_017022	Enterococcus	faecium	Aus0004	Chromosome
NC_017032	Enterococcus	faecium	Aus0004 AUS0004_p1	Plasmid
NC_017023	Enterococcus	faecium	Aus0004 AUS0004_p2	Plasmid
NC_017024	Enterococcus	faecium	Aus0004 AUS0004_p3	Plasmid
NC_021994	Enterococcus	faecium	Aus0085	Chromosome
NC_021987	Enterococcus	faecium	Aus0085 plasmid p1	Plasmid

Continued on next page

Table 1 – continued from previous page

Accession	Genus	Species	Subspecies/strain	Type
NC_021995	Enterococcus	faecium	Aus0085 plasmid p2	Plasmid
NC_021988	Enterococcus	faecium	Aus0085 plasmid p3	Plasmid
NC_021989	Enterococcus	faecium	Aus0085 plasmid p4	Plasmid
NC_021996	Enterococcus	faecium	Aus0085 plasmid p5	Plasmid
NC_021990	Enterococcus	faecium	Aus0085 plasmid p6	Plasmid
NC_017960	Enterococcus	faecium	DO	Chromosome
NC_017961	Enterococcus	faecium	DO plasmid 1	Plasmid
NC_017962	Enterococcus	faecium	DO plasmid 2	Plasmid
NC_020207	Enterococcus	faecium	NRRL B-2354	Chromosome
NC_020208	Enterococcus	faecium	NRRL B-2354 pNB2354_1	Plasmid
NC_017316	Enterococcus	faecium	OG1RF	Chromosome
NC_018081	Enterococcus	hirae	ATCC 9790	Chromosome
NC_015845	Enterococcus	hirae	ATCC 9790 pTG9790	Plasmid
NC_022878	Enterococcus	mundtii	QU 25	Chromosome
NC_022881	Enterococcus	mundtii	QU 25 pQY003	Plasmid
NC_022884	Enterococcus	mundtii	QU 25 pQY024	Plasmid
NC_022880	Enterococcus	mundtii	QU 25 pQY039	Plasmid
NC_022883	Enterococcus	mundtii	QU 25 pQY082	Plasmid
NC_022879	Enterococcus	mundtii	QU 25 pQY182	Plasmid
NC_021023	Enterococcus		sp. 7L76 draft genome	Chromosome
NC_017340	Staphylococcus	aureus	04-02981	Chromosome
NC_018608	Staphylococcus	aureus	08BA02176	Chromosome
NC_021670	Staphylococcus	aureus	Bmb9393	Chromosome
NC_021657	Staphylococcus	aureus	Bmb9393 pBmb9393	Plasmid
NC_021554	Staphylococcus	aureus	CA-347	Chromosome

Continued on next page

Table 1 – continued from previous page

Accession	Genus	Species	Subspecies/strain	Type
NC_021552	Staphylococcus	aureus	CA-347	Plasmid
NC_021059	Staphylococcus	aureus	M1	Chromosome
NC_021060	Staphylococcus	aureus	M1 pSK67-M1	Plasmid
NC_007622	Staphylococcus	aureus	RF122	Chromosome
NC_017351	Staphylococcus	aureus	subsp. aureus 11819-97	Chromosome
NC_017350	Staphylococcus	aureus	subsp. aureus 11819-97 p11819-97	Plasmid
NC_022113	Staphylococcus	aureus	subsp. aureus 55/2053	Chromosome
NC_022126	Staphylococcus	aureus	subsp. aureus 55/2053	Plasmid
NC_022222	Staphylococcus	aureus	subsp. aureus 6850	Chromosome
NC_017673	Staphylococcus	aureus	subsp. aureus 71193	Chromosome
NC_022226	Staphylococcus	aureus	subsp. aureus CN1	Chromosome
NC_022227	Staphylococcus	aureus	subsp. aureus CN1	Plasmid
NC_022228	Staphylococcus	aureus	subsp. aureus CN1	Plasmid
NC_002951	Staphylococcus	aureus	subsp. aureus COL	Chromosome
NC_006629	Staphylococcus	aureus	subsp. aureus COL pT181	Plasmid
NC_017343	Staphylococcus	aureus	subsp. aureus ECT-R 2	Chromosome
NC_017346	Staphylococcus	aureus	subsp. aureus ECT-R 2 pLUH01	Plasmid
NC_017344	Staphylococcus	aureus	subsp. aureus ECT-R 2 pLUH02	Plasmid
NC_017337	Staphylococcus	aureus	subsp. aureus ED133	Chromosome
NC_013450	Staphylococcus	aureus	subsp. aureus ED98	Chromosome
NC_013451	Staphylococcus	aureus	subsp. aureus ED98 pAVY	Plasmid

Continued on next page

Table 1 – continued from previous page

Accession	Genus	Species	Subspecies/strain	Type
NC_017763	Staphylococcus	aureus	subsp. aureus HO 5096 0412	Chromosome
NC_009632	Staphylococcus	aureus	subsp. aureus JH1	Chromosome
NC_009619	Staphylococcus	aureus	subsp. aureus JH1 pSJH101	Plasmid
NC_009477	Staphylococcus	aureus	subsp. aureus JH9 pSJH901	Plasmid
NC_017338	Staphylococcus	aureus	subsp. aureus JKD6159	Chromosome
NC_017339	Staphylococcus	aureus	subsp. aureus JKD6159 pSaa6159	Plasmid
NC_017349	Staphylococcus	aureus	subsp. aureus LGA251	Chromosome
NC_017348	Staphylococcus	aureus	subsp. aureus LGA251 pLGA251	Plasmid
NC_016928	Staphylococcus	aureus	subsp. aureus M013	Chromosome
NC_002952	Staphylococcus	aureus	subsp. aureus MRSA252	Chromosome
NC_016941	Staphylococcus	aureus	subsp. aureus MSHR1132	Chromosome
NC_016942	Staphylococcus	aureus	subsp. aureus MSHR1132 pST75	Plasmid
NC_002953	Staphylococcus	aureus	subsp. aureus MSSA476	Chromosome
NC_009782	Staphylococcus	aureus	subsp. aureus Mu3	Chromosome
NC_002758	Staphylococcus	aureus	subsp. aureus Mu50	Chromosome
NC_002774	Staphylococcus	aureus	subsp. aureus Mu50 VRSAp	Plasmid
NC_003923	Staphylococcus	aureus	subsp. aureus MW2	Chromosome
NC_002745	Staphylococcus	aureus	subsp. aureus N315	Chromosome

Continued on next page

Table 1 – continued from previous page

Accession	Genus	Species	Subspecies/strain	Type
NC_003140	Staphylococcus	aureus	subsp. aureus N315 pN315	Plasmid
NC_007795	Staphylococcus	aureus	subsp. aureus NCTC 8325	Chromosome
NC_022443	Staphylococcus	aureus	subsp. aureus SA40	Chromosome
NC_022442	Staphylococcus	aureus	subsp. aureus SA957	Chromosome
NC_02053	Staphylococcus	aureus	subsp. aureus ST228	Chromosome
NC_020529	Staphylococcus	aureus	subsp. aureus ST228	Chromosome
NC_020532	Staphylococcus	aureus	subsp. aureus ST228	Chromosome
NC_020533	Staphylococcus	aureus	subsp. aureus ST228	Chromosome
NC_020536	Staphylococcus	aureus	subsp. aureus ST228	Chromosome
NC_020537	Staphylococcus	aureus	subsp. aureus ST228	Chromosome
NC_020564	Staphylococcus	aureus	subsp. aureus ST228	Chromosome
NC_020566	Staphylococcus	aureus	subsp. aureus ST228	Chromosome
NC_020568	Staphylococcus	aureus	subsp. aureus ST228	Chromosome
NC_020530	Staphylococcus	aureus	subsp. aureus ST228 pI1T1	Plasmid
NC_020531	Staphylococcus	aureus	subsp. aureus ST228 pI1T2	Plasmid
NC_020534	Staphylococcus	aureus	subsp. aureus ST228 pI1T8	Plasmid
NC_020565	Staphylococcus	aureus	subsp. aureus ST228 pI3T3	Plasmid

Continued on next page

Table 1 – continued from previous page

Accession	Genus	Species	Subspecies/strain	Type
NC_020535	Staphylococcus	aureus	subsp. aureus ST228 pI5S5	Plasmid
NC_020567	Staphylococcus	aureus	subsp. aureus ST228 pI6T6	Plasmid
NC_020538	Staphylococcus	aureus	subsp. aureus ST228 pI7S6	Plasmid
NC_020539	Staphylococcus	aureus	subsp. aureus ST228 pI8T7	Plasmid
NC_017333	Staphylococcus	aureus	subsp. aureus ST398	Chromosome
NC_017334	Staphylococcus	aureus	subsp. aureus ST398 pS0385-1	Plasmid
NC_017335	Staphylococcus	aureus	subsp. aureus ST398 pS0385-2	Plasmid
NC_017336	Staphylococcus	aureus	subsp. aureus ST398 pS0385-3	Plasmid
NC_017341	Staphylococcus	aureus	subsp. aureus str. JKD6008	Chromosome
NC_009641	Staphylococcus	aureus	subsp. aureus str. Newman	Chromosome
NC_017347	Staphylococcus	aureus	subsp. aureus T0131	Chromosome
NC_017342	Staphylococcus	aureus	subsp. aureus TCH60	Chromosome
NC_017331	Staphylococcus	aureus	subsp. aureus TW20	Chromosome
NC_017352	Staphylococcus	aureus	subsp. aureus TW20 pTW20_1	Plasmid
NC_017332	Staphylococcus	aureus	subsp. aureus TW20 pTW20_2	Plasmid
NC_007793	Staphylococcus	aureus	subsp. aureus USA300_FPR3757	Chromosome

Continued on next page

Table 1 – continued from previous page

Accession	Genus	Species	Subspecies/strain	Type
NC_007790	Staphylococcus	aureus	subsp. aureus USA300_FPR3757 pUSA01	Plasmid
NC_007791	Staphylococcus	aureus	subsp. aureus USA300_FPR3757 pUSA02	Plasmid
NC_007792	Staphylococcus	aureus	subsp. aureus USA300_FPR3757 pUSA03	Plasmid
NC_010079	Staphylococcus	aureus	subsp. aureus USA300_TCH1516	Chromosome
NC_012417	Staphylococcus	aureus	subsp. aureus USA300_TCH1516 pUSA01-HOU	Plasmid
NC_010063	Staphylococcus	aureus	subsp. aureus USA300_TCH1516 pUSA300HOUMR	Plasmid
NC_016912	Staphylococcus	aureus	subsp. aureus VC40	Chromosome
NC_022604	Staphylococcus	aureus	subsp. aureus Z172	Chromosome
NC_022610	Staphylococcus	aureus	subsp. aureus Z172 pZ172_1	Plasmid
NC_022605	Staphylococcus	aureus	subsp. aureus Z172 pZ172_2	Plasmid
NC_012121	Staphylococcus	carnosus	subsp. carnosus TM300	Chromosome
NC_004461	Staphylococcus	epidermidis	ATCC 12228	Chromosome
NC_005008	Staphylococcus	epidermidis	ATCC 12228 pSE-12228-01	Plasmid
NC_005007	Staphylococcus	epidermidis	ATCC 12228 pSE-12228-02	Plasmid
NC_005005	Staphylococcus	epidermidis	ATCC 12228 pSE-12228-04	Plasmid

Continued on next page

Table 1 – continued from previous page

Accession	Genus	Species	Subspecies/strain	Type
NC_005004	Staphylococcus	epidermidis	ATCC 12228 pSE-12228-05	Plasmid
NC_005003	Staphylococcus	epidermidis	ATCC 12228 pSE-12228-06	Plasmid
NC_002976	Staphylococcus	epidermidis	RP62A	Chromosome
NC_006663	Staphylococcus	epidermidis	RP62A pSERP	Plasmid
NC_007168	Staphylococcus	haemolyticus	JCSC1435	Chromosome
NC_007169	Staphylococcus	haemolyticus	JCSC1435 pSHaeA	Plasmid
NC_007170	Staphylococcus	haemolyticus	JCSC1435 pSHaeB	Plasmid
NC_007171	Staphylococcus	haemolyticus	JCSC1435 pSHaeC	Plasmid
NC_013893	Staphylococcus	lugdunensis	HKU09-01	Chromosome
NC_017353	Staphylococcus	lugdunensis	N920143	Chromosome
NC_014925	Staphylococcus	pseudintermedius	HKU10-03	Chromosome
NC_007350	Staphylococcus	saprophyticus	subsp. saprophyticus ATCC 15305	Chromosome
NC_007351	Staphylococcus	saprophyticus	subsp. saprophyticus ATCC 15305 pSSP1	Plasmid
NC_007352	Staphylococcus	saprophyticus	subsp. saprophyticus ATCC 15305 pSSP2	Plasmid
NC_020164	Staphylococcus	warneri	SG1	Chromosome
NC_020274	Staphylococcus	warneri	SG1 clone pvSw1	Plasmid
NC_020264	Staphylococcus	warneri	SG1 clone pvSw2	Plasmid
NC_020265	Staphylococcus	warneri	SG1 clone pvSw3	Plasmid
NC_020266	Staphylococcus	warneri	SG1 clone pvSw4	Plasmid
NC_020267	Staphylococcus	warneri	SG1 clone pvSw5	Plasmid
NC_020268	Staphylococcus	warneri	SG1 clone pvSw6	Plasmid
NC_020269	Staphylococcus	warneri	SG1 clone pvSw7	Plasmid
NC_020165	Staphylococcus	warneri	SG1 pSZ4	Plasmid

Appendix B: Further conservation analyses

.1 Staphylococcus aureus subsp. aureus N315

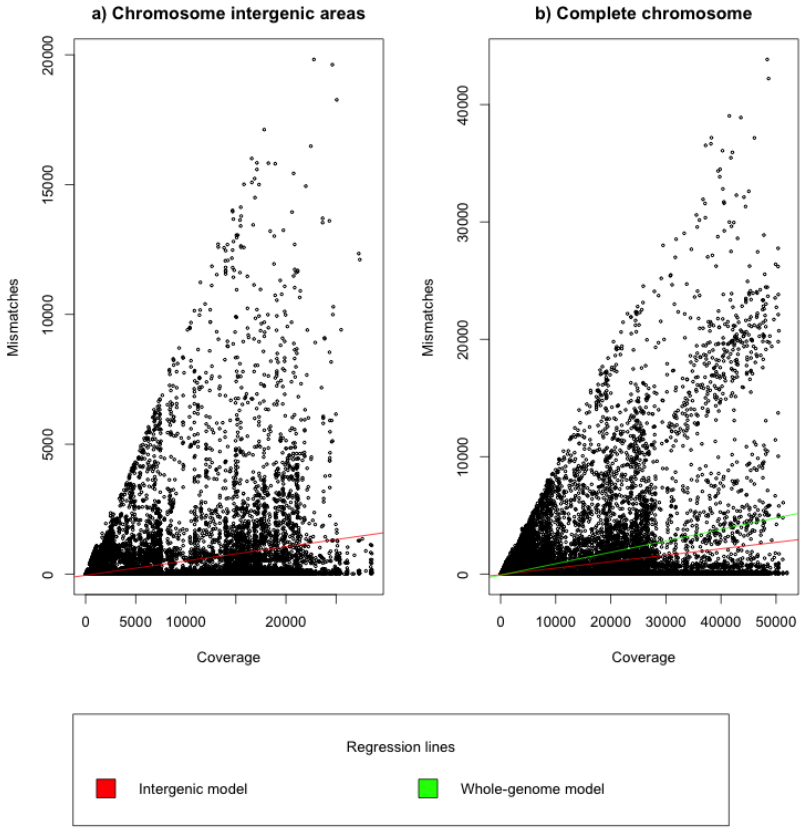

Figure 1: The left figure shows the mismatches vs. coverage for the intergenic areas of the *Staphylococcus aureus subsp aureus N315* chromosome, with a red regression line indicating the expected average number of mismatches. The right figure shows data from the whole *S. aureus subsp aureus N315* chromosome, with a red regression line indicating the expected average number of mismatches based on the intergenic data, and a green regression line indicating the average number of mismatches when the whole chromosome BLAST result is used as data for the model.

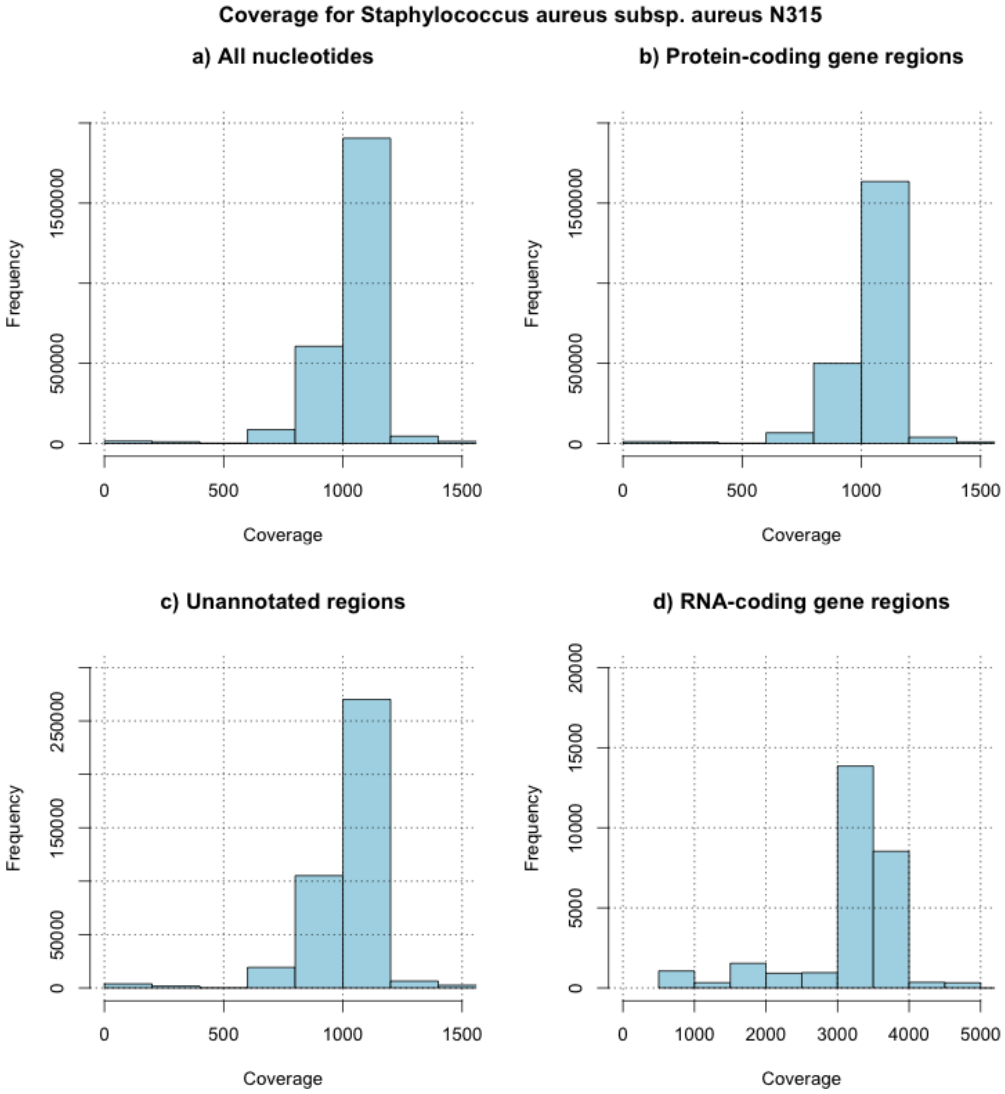

Figure 2: Coverage distributions for the nucleic positions in the *Staphylococcus aureus subsp aureus N315* chromosome data. a) shows the distribution of coverages for the complete chromosome, while b), c) and d) shows coverage distributions for protein coding- (i.e. the regions with annotations for protein-coding genes), un-annotated- and RNA-coding chromosome regions respectively. Notice that the frequencies shown in a) and b) are higher than in c) and especially d). This stems from the fact that the majority of the chromosome regions are annotated for protein-coding genes.

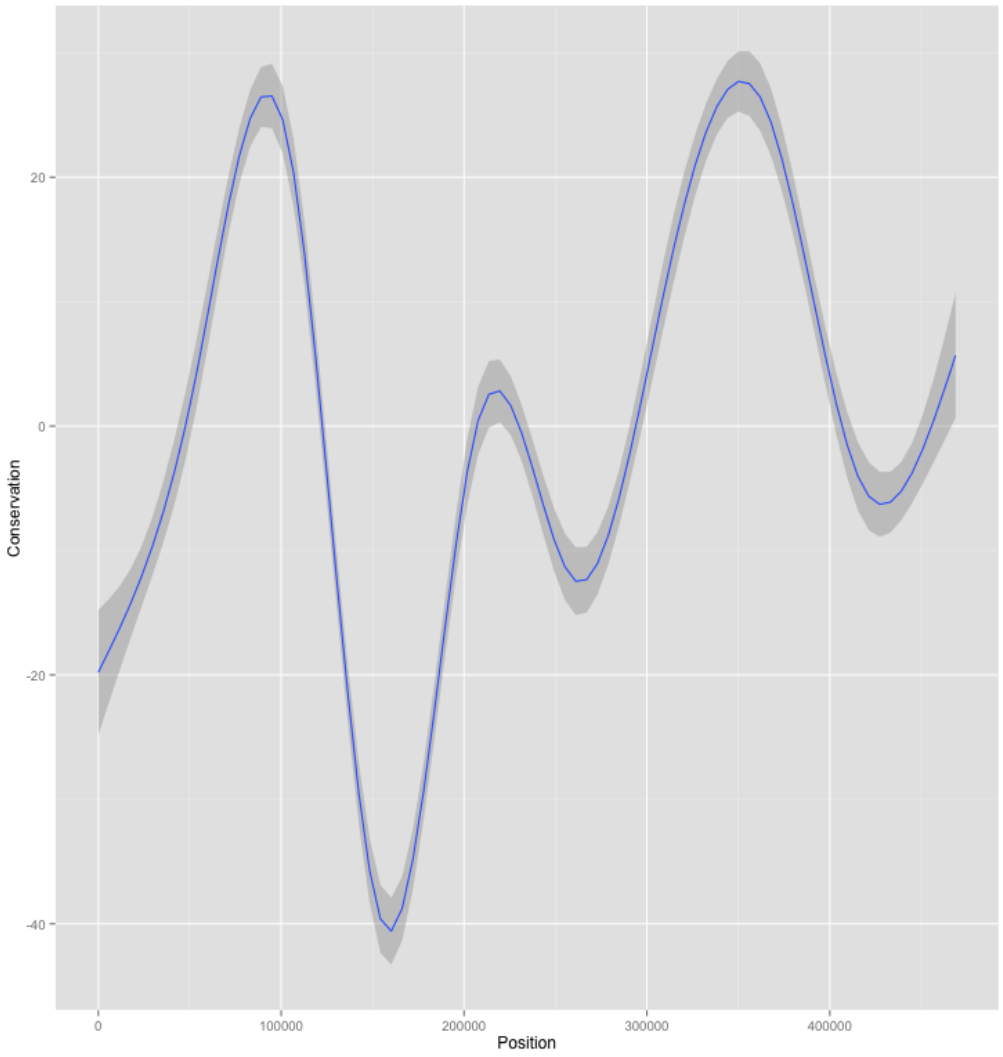

Figure 3: Shows the mean conservation by position in the *Staphylococcus aureus subsp. aureus N315* intergenic data. A 95% confidence interval is shown around the mean line. The figure is generated with the ggplot2-package in R using *stat_smooth* (http://docs.ggplot2.org/0.9.3.1/stat_smooth.html). Default arguments are used, and for datasets with 1000 or more observations like this, the default smoothing model is GAM (http://www.inside-r.org/r-doc/mgcv/gam)

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = intergenicModel_Conservation ~ Annotations, data = wholegenome_positionInfo)

$Annotations
                                diff         lwr          upr     p adj
RNA-coding-Protein-coding    128.096900  123.734847  132.4589523 0.0000000
Unannotated-Protein-coding    -1.954721   -3.150994   -0.7584478 0.0003774
Unannotated-RNA-coding      -130.051620 -134.524436 -125.5788045 0.0000000
```

Figure 4: Printout from the *TukeyHSD*-function in R (http://stat.ethz.ch/R-manual/R-patched/library/stats/html/TukeyHSD.html) when used on the *Staphylococcus aureus subsp. aureus N315* conservation data. Shows the differences between mean conservation value of the groups: Protein coding gene annotation (*Protein-coding* in printout), RNA-coding gene annotation (*RNA-coding* in printout) and unannotated regions (*Unannotated* in printout) under *diff*. Default confidence level is 95%. The statistical model used to predict the conservation values is built on the intergenic data only.

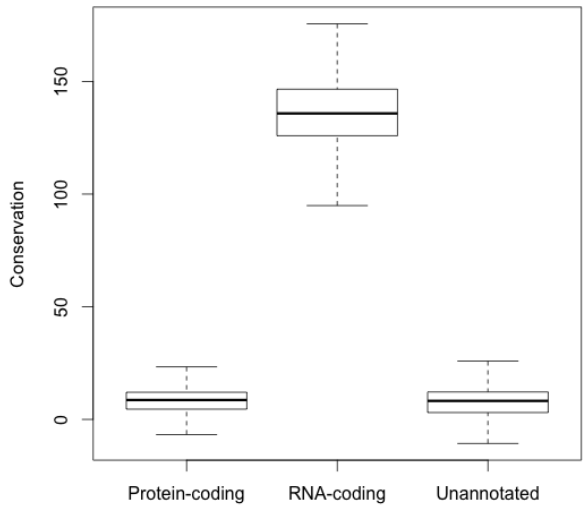

Figure 5: Box- and whiskers plot of the conservational values for the nucleic positions in each of the three groups: *DNA annotation*, *RNA annotation* and *Unannotated*. Data is from the orfstat output for the bacterium *Staphylococcus aureus subsp. aureus N315*. For each group the black bolded line is the median, the horizontal lines under and over the median are the first and third quartiles, repectively. 50 % of the data points are inside the boundaries of this box. R is used to generate the plot, http://stat.ethz.ch/R-manual/R-patched/library/graphics/html/boxplot.html. The statistical model used to predict the conservation values is built on the intergenic data only.

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Conservation ~ Annotations, data = wholegenome_positionInfo)

$Annotations
                                  diff         lwr        upr p adj
RNA-coding-Protein-coding   218.262315  214.011557  222.51307     0
Unannotated-Protein-coding    9.134377    7.968626   10.30013     0
Unannotated-RNA-coding     -209.127938 -213.486634 -204.76924     0
```

Figure 6: Shows the same information as in Figure 4, except the whole chromosome data for *Staphylococcus aureus subsp. aureus N315* is used to estimate the parameters for the statistical model.

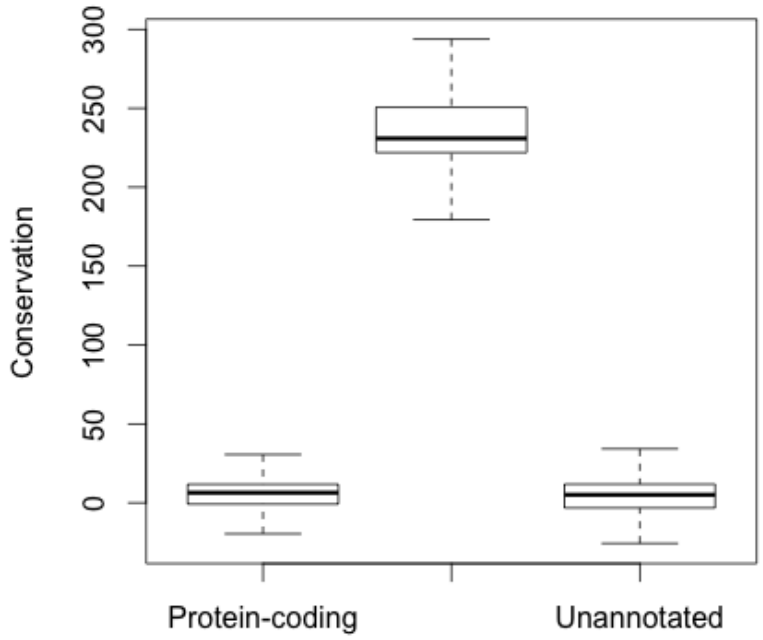

Figure 7: Box- and whiskers plot showing the same three groups as in Figure 5. The description from Figure 5 applies here, except the full chromosome data from *Staphylococcus aureus subsp. aureus N315* is used to build the statistical model, instead of only the intergenic regions.

.2 Enterococcus faecium NRRL B-2354

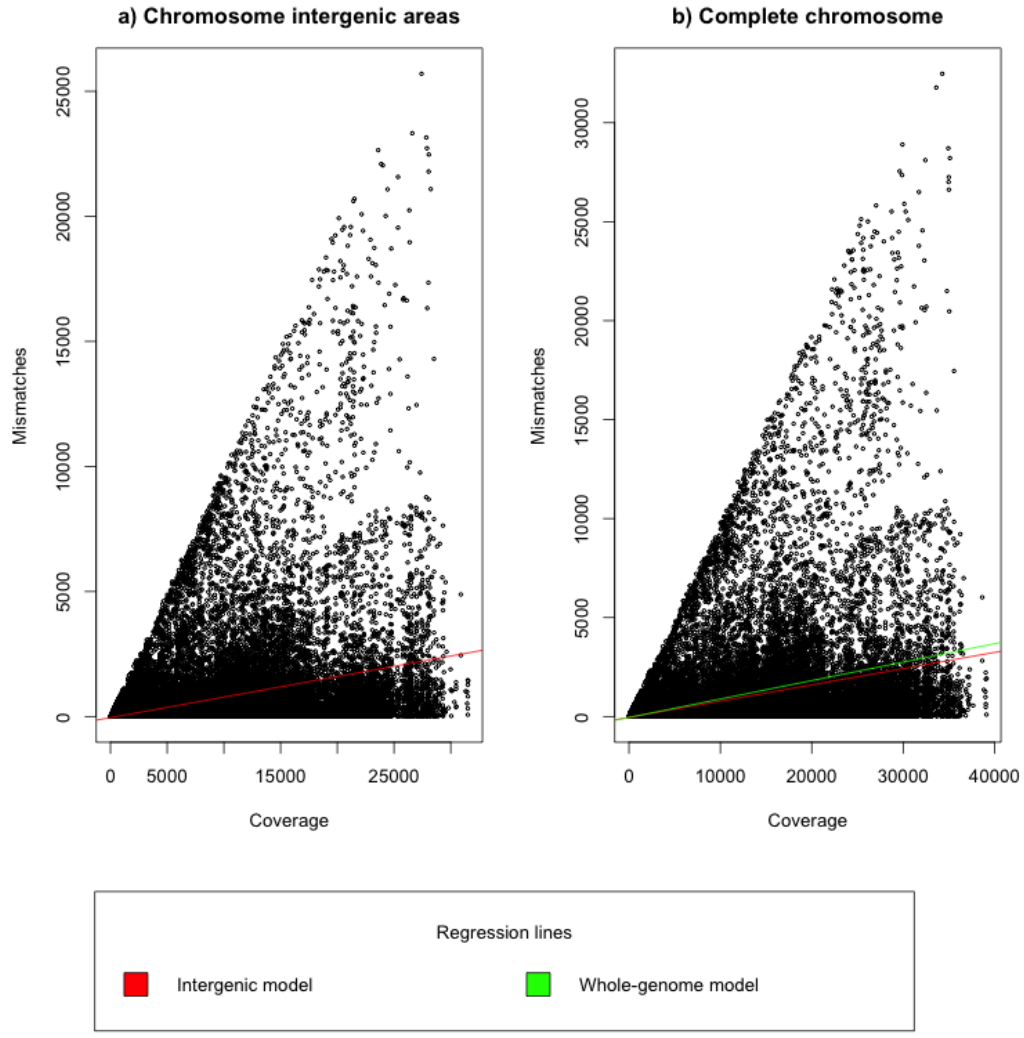

Figure 8: The left figure shows the mismatches vs. coverage for the intergenic areas of the *Enterococcus faecium NRRL B-2354* chromosome, with a red regression line indicating the expected average number of mismatches. The right figure shows data from the whole *E. faecium NRRL B-2354* chromosome, with a red regression line indicating the expected average number of mismatches based on the intergenic data, and a green regression line indicating the average number of mismatches when the whole chromosome BLAST result is used as data for the model.

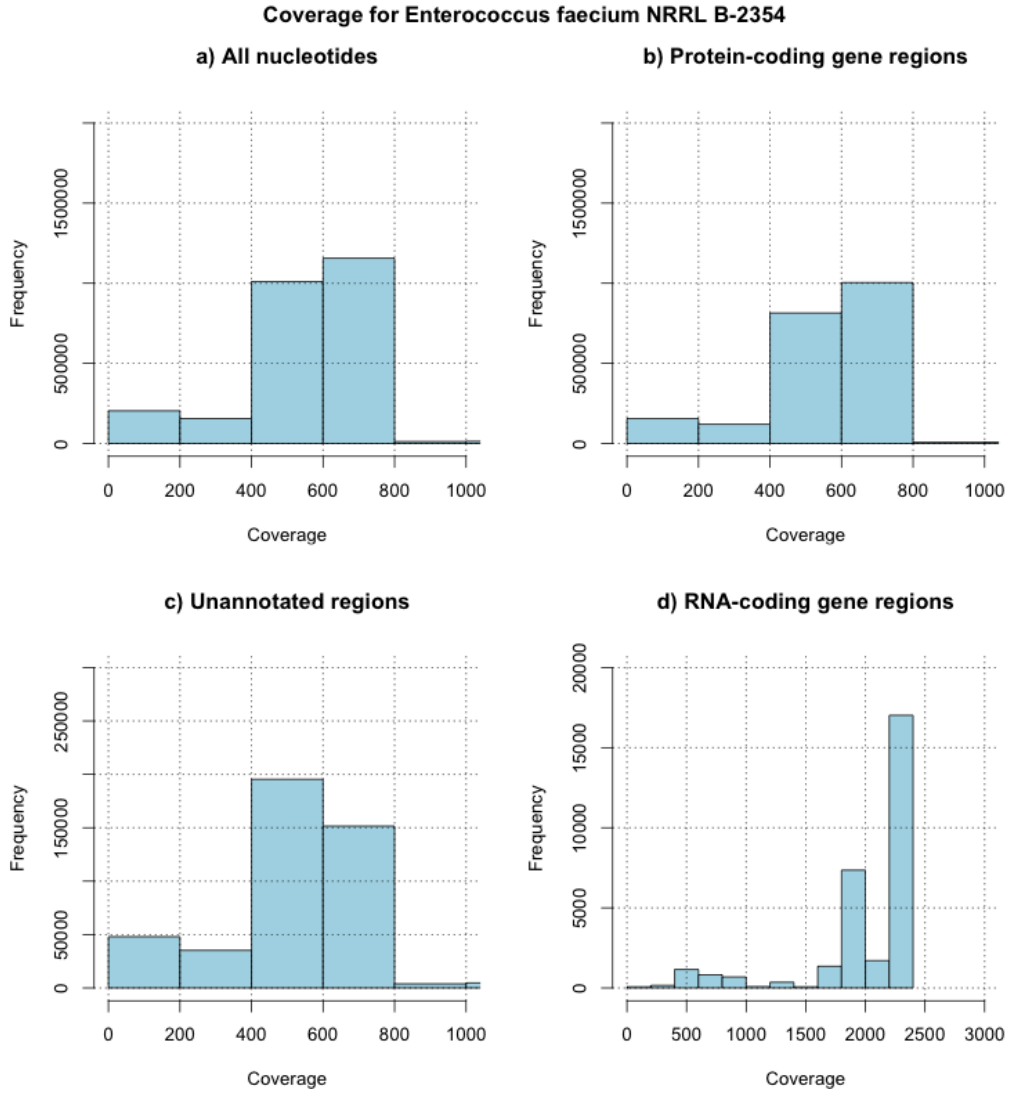

Figure 9: Coverage distributions for the nucleic positions in the *Enterococcus faecium NRRL B-2354* chromosome data. a) shows the distribution of coverages for the complete chromosome, while b), c) and d) shows coverage distributions for protein coding- (i.e. the regions with annotations for protein-coding genes), un-annotated- and RNA-coding chromosome regions respectively. Notice that the frequencies shown in a) and b) are higher than in c) and especially d). This stems from the fact that the majority of the chromosome regions are annotated for protein-coding genes.

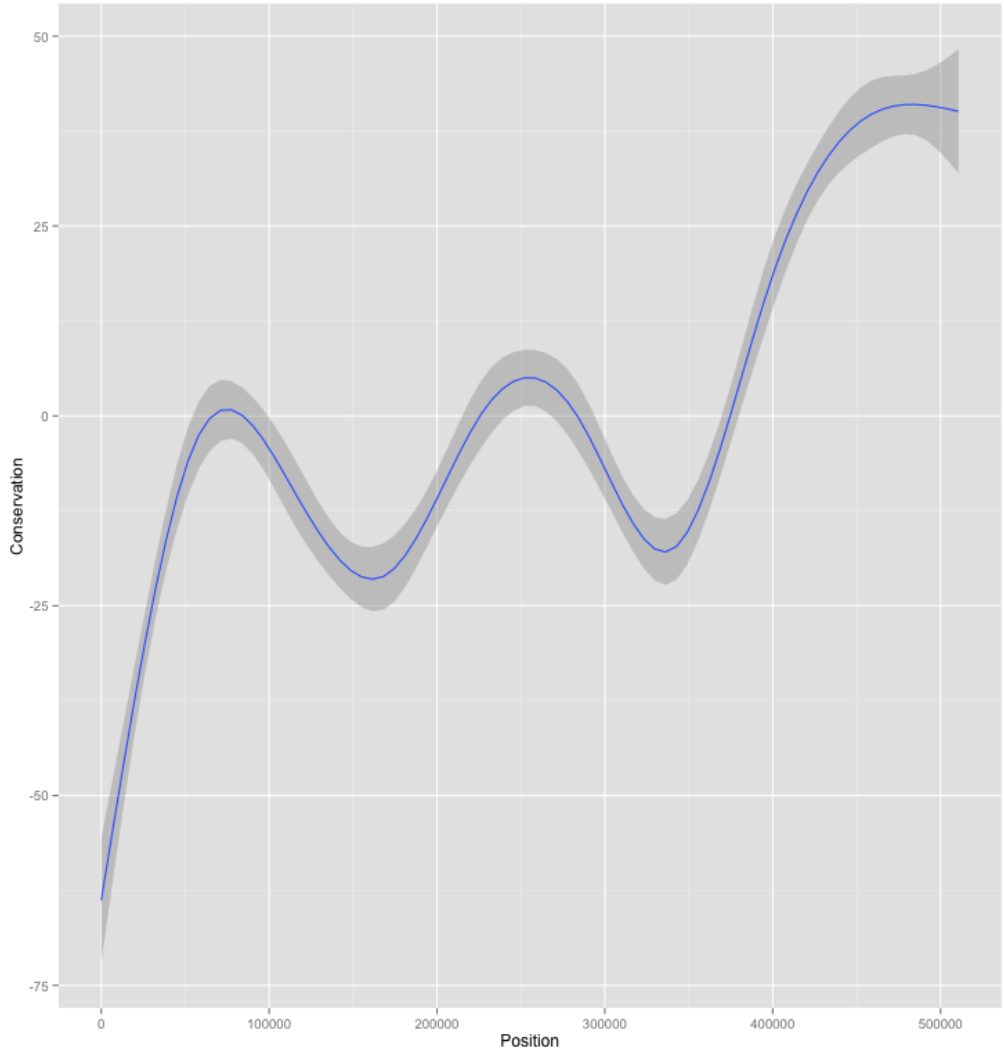

Figure 10: Shows the mean conservation by position in the *Enterococcus faecium NRRL B-2354* intergenic data. A 95% confidence interval is shown around the mean line. The figure is generated with the ggplot2-package in R using *stat_smooth* (http://docs.ggplot2.org/0.9.3.1/stat_smooth.html). Default arguments are used, and for datasets with 1000 or more observations like this, the default smoothing model is GAM (http://www.inside-r.org/r-doc/mgcv/gam)

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = intergenicModel_Conservation ~ Annotations, data = wholegenome_positionInfo)

$Annotations
                               diff        lwr        upr p adj
RNA-coding-Protein-coding   121.8111  117.99248  125.62968     0
Unannotated-Protein-coding  -13.9433  -15.00768  -12.87893     0
Unannotated-RNA-coding     -135.7544 -139.66555 -131.84323     0
```

Figure 11: Printout from the *TukeyHSD*-function in R (http://stat.ethz.ch/R-manual/R-patched/library/stats/html/TukeyHSD.html) when used on the *Enterococcus faecium NRRL B-2354* conservation data. Shows the differences between mean conservation value of the groups: Protein coding gene annotation (*Protein-coding* in printout), RNA-coding gene annotation (*RNA-coding* in printout) and unannotated regions (*Unannotated* in printout) under *diff*. Default confidence level is 95%. The statistical model used to predict the conservation values is built on the intergenic data only.

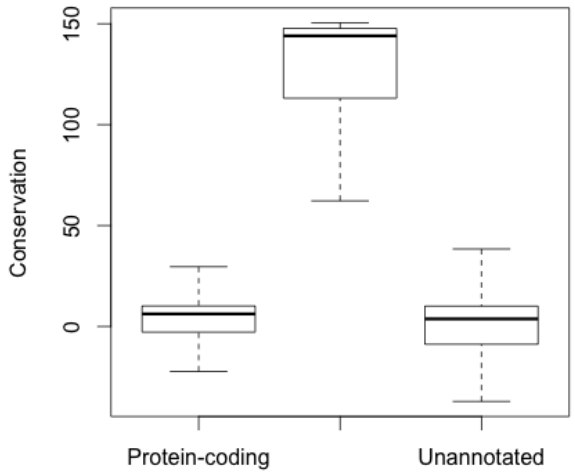

Figure 12: Box- and whiskers plot of the conservational values for the nucleic positions in each of the three groups: *DNA annotation*, *RNA annotation* and *Unannotated*. Data is from the orfstat output for the bacterium *Enterococcus faecium NRRL B-2354*. For each group the black bolded line is the median, the horizontal lines under and over the median are the first and third quartiles, repectively. 50 % of the data points are inside the boundaries of this box. R is used to generate the plot, http://stat.ethz.ch/R-manual/R-patched/library/graphics/html/boxplot.html. The statistical model used to predict the conservation values is built on the intergenic data only.

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Conservation ~ Annotations, data = wholegenome_positionInfo)

$Annotations
                                  diff         lwr        upr p adj
RNA-coding-Protein-coding   137.71437  133.904372  141.52437     0
Unannotated-Protein-coding   -5.68154   -6.743519   -4.61956     0
Unannotated-RNA-coding     -143.39591 -147.298264 -139.49356     0
```

Figure 13: Shows the same information as in Figure 11, except the whole chromosome data for *Enterococcus faecium NRRL B-2354* is used to estimate the parameters for the statistical model.

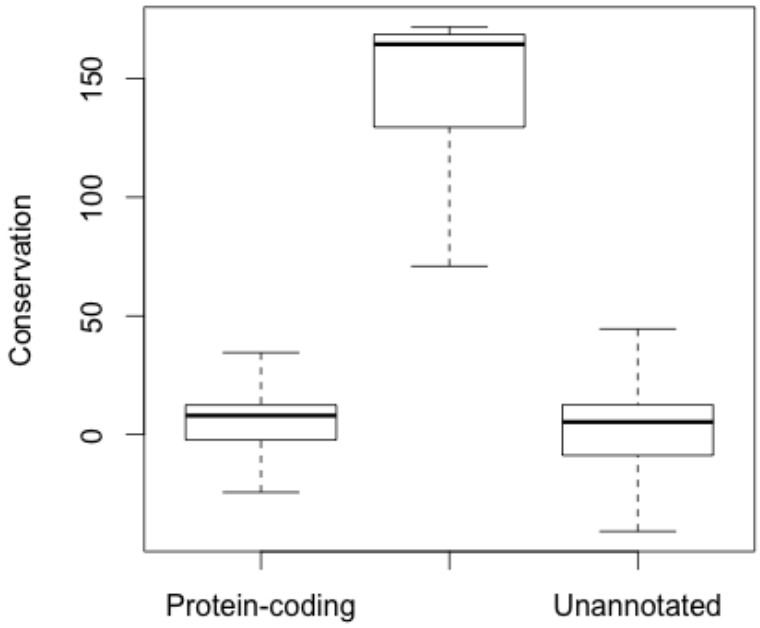

Figure 14: Box- and whiskers plot showing the same three groups as in Figure 12. The description from Figure 12 applies here, except the full chromosome data from *Enterococcus faecium NRRL B-2354* is used to build the statistical model, instead of only the intergenic regions.

.3 Enterococcus faecium Aus0004

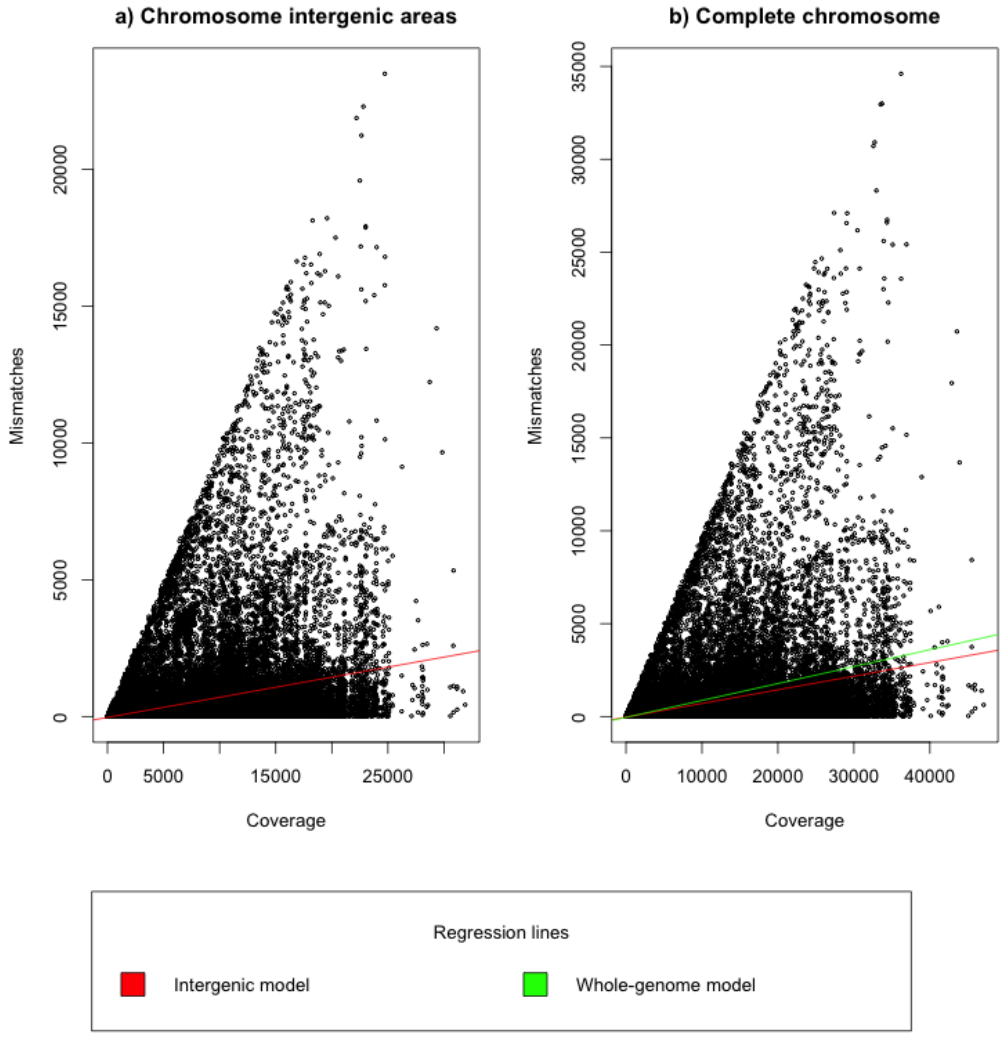

Figure 15: The left figure shows the mismatches vs. coverage for the intergenic areas of the *Enterococcus faecium Aus0004* chromosome, with a red regression line indicating the expected average number of mismatches. The right figure shows data from the whole *E. faecium Aus0004* chromosome, with a red regression line indicating the expected average number of mismatches based on the intergenic data, and a green regression line indicating the average number of mismatches when the whole chromosome BLAST result is used as data for the model.

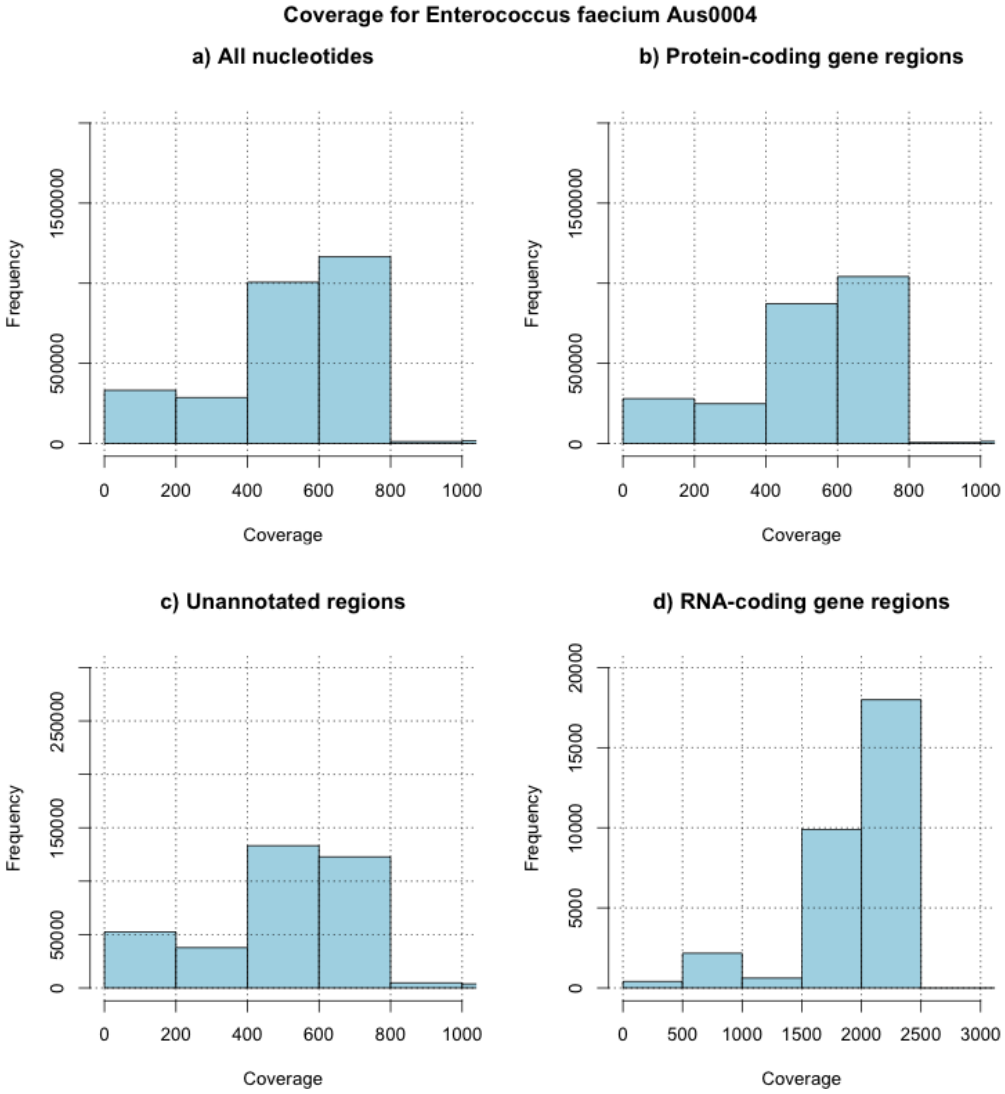

Figure 16: Coverage distributions for the nucleic positions in the *E. faecium Aus0004* chromosome data. a) shows the distribution of coverages for the complete chromosome, while b), c) and d) shows coverage distributions for protein coding- (i.e. the regions with annotations for protein-coding genes), un-annotated- and RNA-coding chromosome regions respectively. Notice that the frequencies shown in a) and b) are higher than in c) and especially d). This stems from the fact that the majority of the chromosome regions are annotated for protein-coding genes.

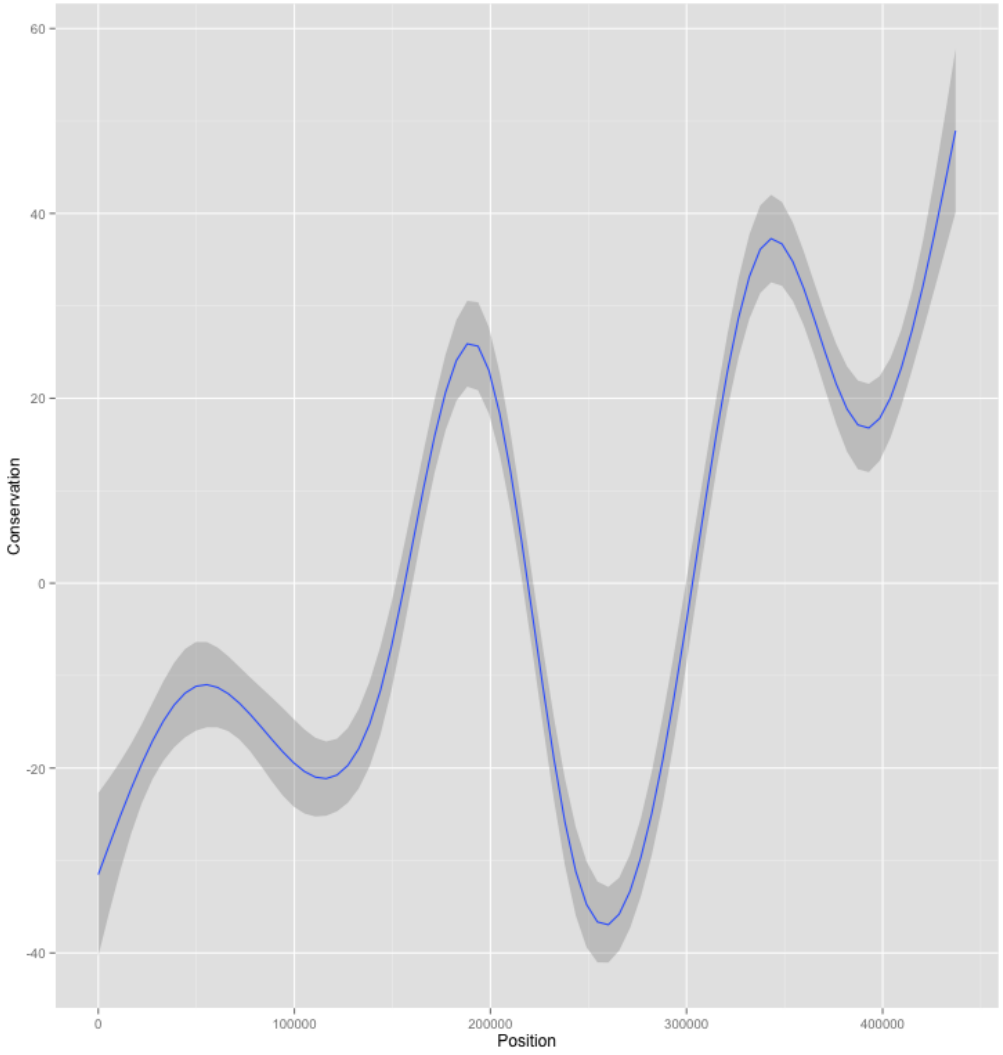

Figure 17: Shows the mean conservation by position in the *Enterococcus faecium Aus0004* intergenic data. A 95% confidence interval is shown around the mean line. The figure is generated with the ggplot2-package in R using *stat_smooth* (http://docs.ggplot2.org/0.9.3.1/stat_smooth.html). Default arguments are used, and for datasets with 1000 or more observations like this, the default smoothing model is GAM (http://www.inside-r.org/r-doc/mgcv/gam)

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = intergenicModel_Conservation ~ Annotations, data = wholegenome_positionInfo)

$Annotations
                                 diff        lwr        upr p adj
RNA-coding-Protein-coding     107.30975  103.80359  110.81590     0
Unannotated-Protein-coding    -19.88109  -20.92005  -18.84213     0
Unannotated-RNA-coding       -127.19083 -130.80646 -123.57521     0
```

Figure 18: Printout from the *TukeyHSD*-function in R (http://stat.ethz.ch/R-manual/R-patched/library/stats/html/TukeyHSD.html) when used on the *Enterococcus faecium Aus0004* conservation data. Shows the differences between mean conservation value of the groups: Protein coding gene annotation (*Protein-coding* in printout), RNA-coding gene annotation (*RNA-coding* in printout) and unannotated regions (*Unannotated* in printout) under *diff*. Default confidence level is 95%. The statistical model used to predict the conservation values is built on the intergenic data only.

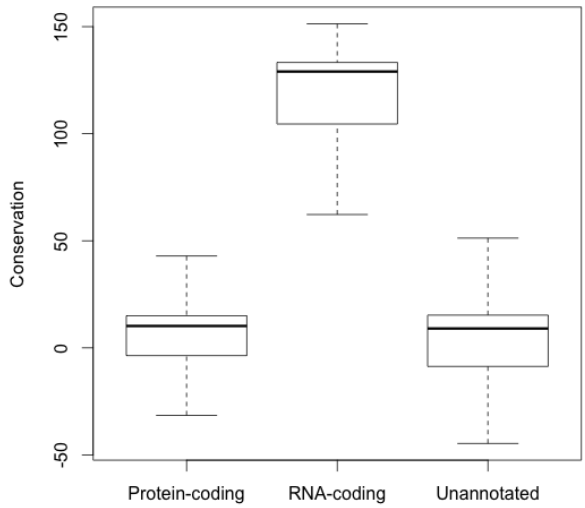

Figure 19: Box- and whiskers plot of the conservational values for the nucleic positions in each of the three groups: *DNA annotation*, *RNA annotation* and *Unannotated*. Data is from the orfstat output for the bacterium *Enterococcus faecium Aus0004*. For each group the black bolded line is the median, the horizontal lines under and over the median are the first and third quartiles, repectively. 50 % of the data points are inside the boundaries of this box. R is used to generate the plot, http://stat.ethz.ch/R-manual/R-patched/library/graphics/html/boxplot.html. The statistical model used to predict the conservation values is built on the intergenic data only.

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Conservation ~ Annotations, data = wholegenome_positionInfo)

$Annotations
                                  diff         lwr         upr p adj
RNA-coding-Protein-coding   132.131507  128.644261  135.618752     0
Unannotated-Protein-coding   -3.409548   -4.442903   -2.376193     0
Unannotated-RNA-coding     -135.541054 -139.137177 -131.944932     0
```

Figure 20: Shows the same information as in Figure 18, except the whole chromosome data for *Enterococcus faecium Aus0004* is used to estimate the parameters for the statistical model.

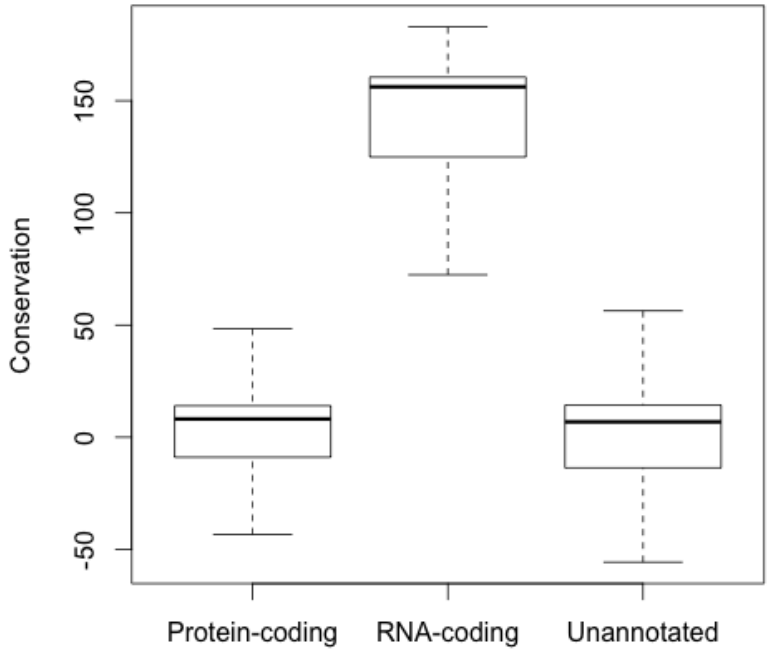

Figure 21: Box- and whiskers plot showing the same three groups as in Figure 19. The description from Figure 19 applies here, except the full chromosome data from *Enterococcus faecium Aus0004* is used to build the statistical model, instead of only the intergenic regions.

.4 Staphylococcus aureus LGA251

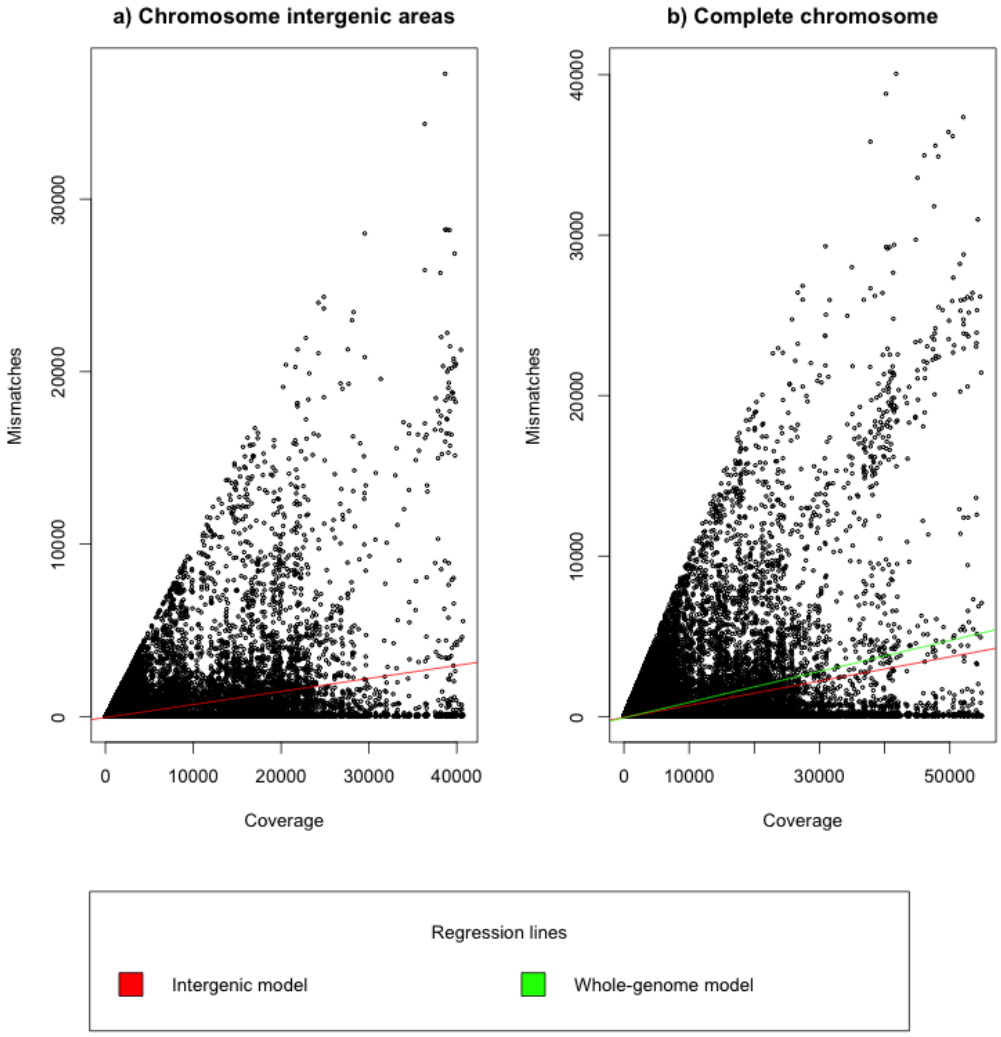

Figure 22: The left figure shows the mismatches vs. coverage for the intergenic areas of the *Staphylococcus aureus LGA251* chromosome, with a red regression line indicating the expected average number of mismatches. The right figure shows data from the whole *S. aureus LGA251* chromosome, with a red regression line indicating the expected average number of mismatches based on the intergenic data, and a green regression line indicating the average number of mismatches when the whole chromosome BLAST result is used as data for the model.

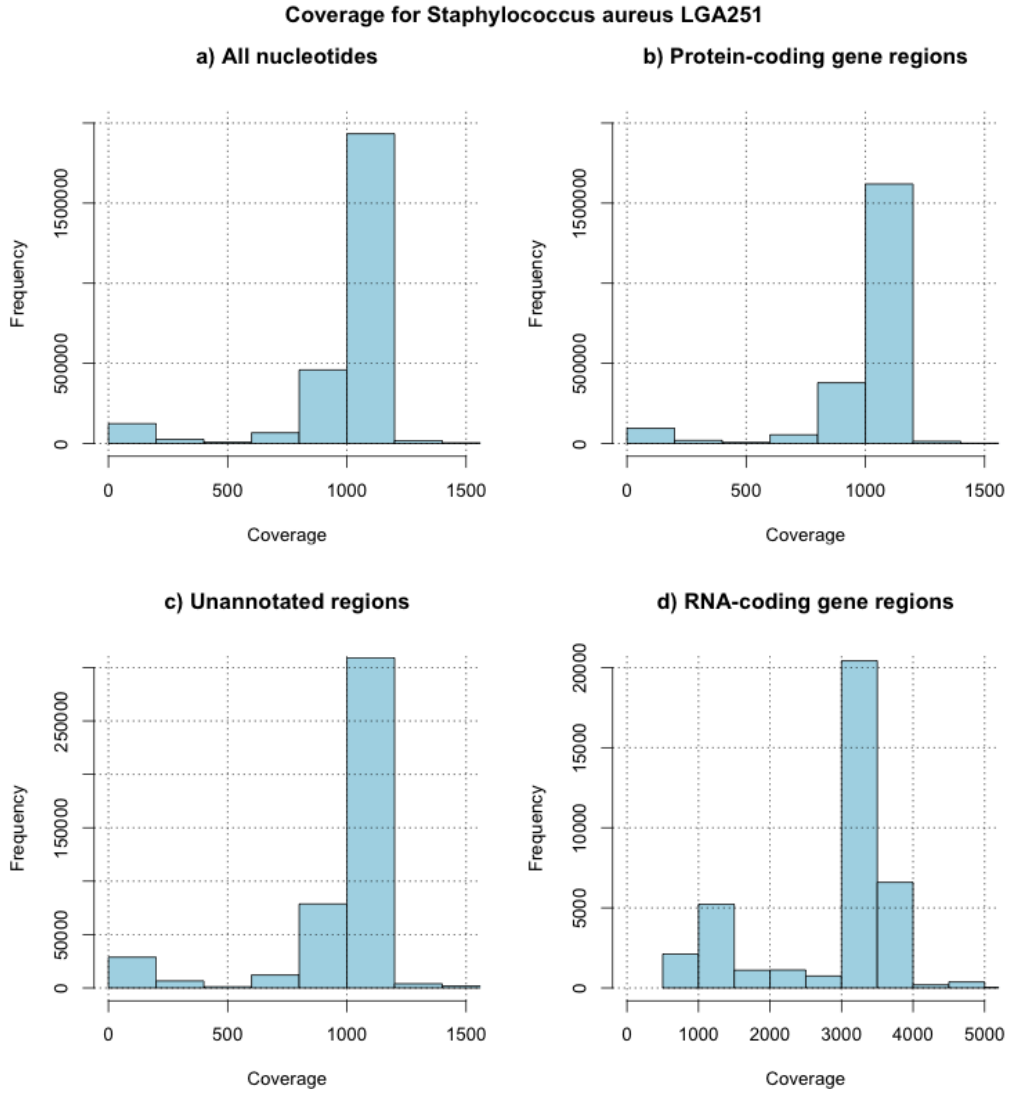

Figure 23: Coverage distributions for the nucleic positions in the *Staphylococcus aureus LGA251* chromosome data. a) shows the distribution of coverages for the complete chromosome, while b), c) and d) shows coverage distributions for protein coding- (i.e. the regions with annotations for protein-coding genes), un-annotated- and RNA-coding chromosome regions respectively. Notice that the frequencies shown in a) and b) are higher than in c) and especially d). This stems from the fact that the majority of the chromosome regions are annotated for protein-coding genes.

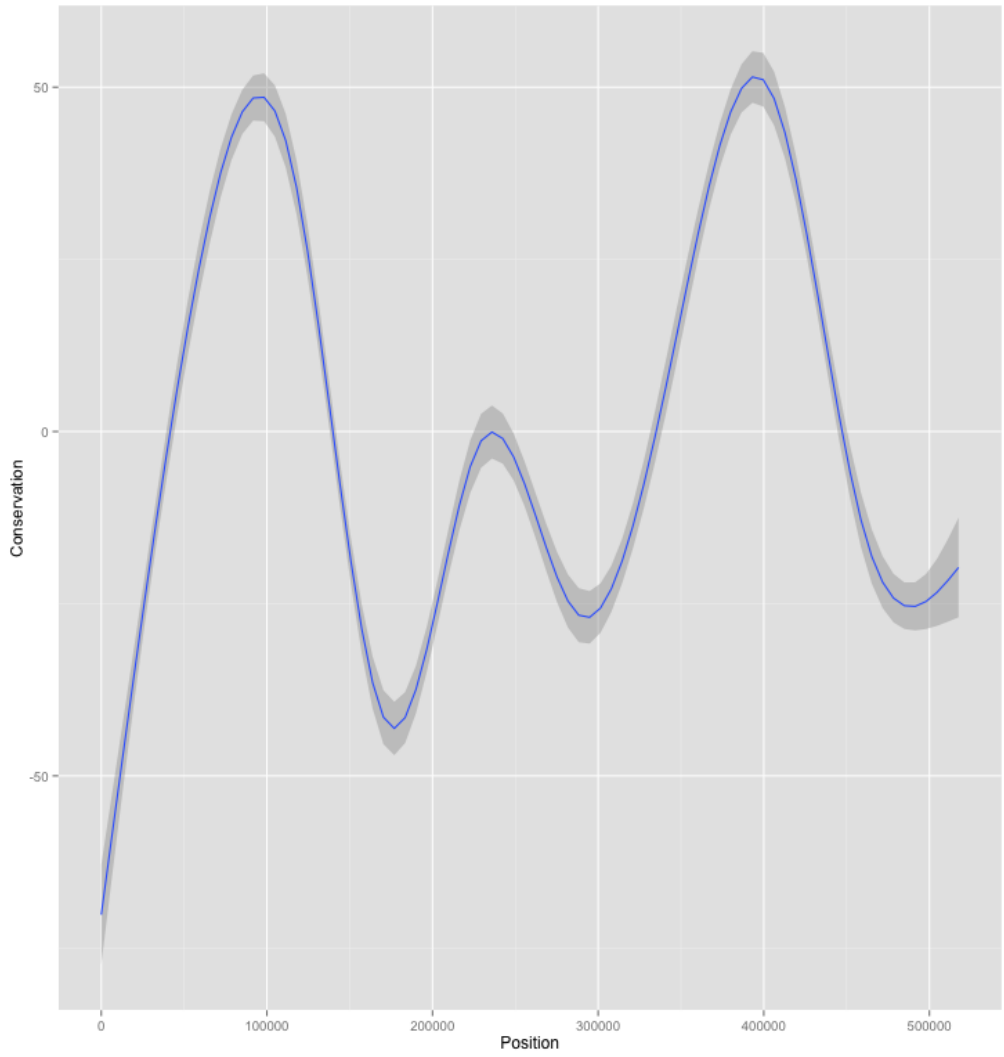

Figure 24: Shows the mean conservation by position in the *Staphylococcus aureus LGA251* intergenic data. A 95% confidence interval is shown around the mean line. The figure is generated with the ggplot2-package in R using *stat_smooth* (http://docs.ggplot2.org/0.9.3.1/stat_smooth.html). Default arguments are used, and for datasets with 1000 or more observations like this, the default smoothing model is GAM (http://www.inside-r.org/r-doc/mgcv/gam)

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = intergenicModel_Conservation ~ Annotations, data = wholegenome_positionInfo)

$Annotations
                                diff        lwr        upr p adj
RNA-coding-Protein-coding     157.98825  154.33819  161.63831     0
Unannotated-Protein-coding    -12.68498  -13.81313  -11.55683     0
Unannotated-RNA-coding       -170.67323 -174.43434 -166.91213     0
```

Figure 25: Printout from the *TukeyHSD*-function in R (http://stat.ethz.ch/R-manual/R-patched/library/stats/html/TukeyHSD.html) when used on the *Staphylococcus aureus LGA251* conservation data. Shows the differences between mean conservation value of the groups: Protein coding gene annotation (*Protein-coding* in printout), RNA-coding gene annotation (*RNA-coding* in printout) and unannotated regions (*Unannotated* in printout) under *diff*. Default confidence level is 95%. The statistical model used to predict the conservation values is built on the intergenic data only.

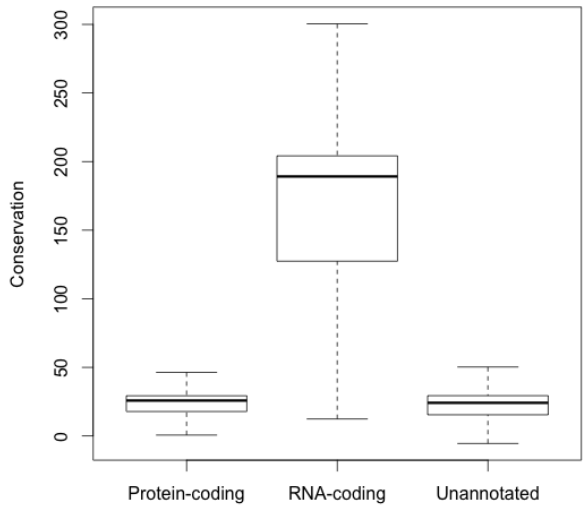

Figure 26: Box- and whiskers plot of the conservational values for the nucleic positions in each of the three groups: *DNA annotation*, *RNA annotation* and *Unannotated*. Data is from the orfstat output for the bacterium *Staphylococcus aureus LGA251*. For each group the black bolded line is the median, the horizontal lines under and over the median are the first and third quartiles, repectively. 50 % of the data points are inside the boundaries of this box. R is used to generate the plot, http://stat.ethz.ch/R-manual/R-patched/library/graphics/html/boxplot.html. The statistical model used to predict the conservation values is built on the intergenic data only.

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Conservation ~ Annotations, data = wholegenome_positionInfo)

$Annotations
                                 diff         lwr         upr p adj
RNA-coding-Protein-coding   196.186859  192.558217  199.815500     0
Unannotated-Protein-coding   -5.053895   -6.175425   -3.932366     0
Unannotated-RNA-coding     -201.240754 -204.979786 -197.501723     0
```

Figure 27: Shows the same information as in Figure 25, except the whole chromosome data for *Staphylococcus aureus LGA251* is used to estimate the parameters for the statistical model.

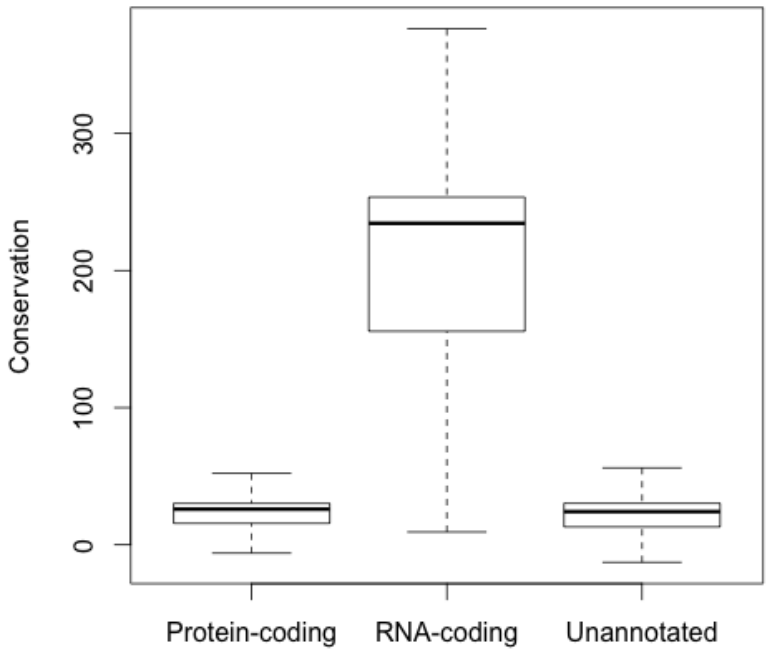

Figure 28: Box- and whiskers plot showing the same three groups as in Figure 26. The description from Figure 26 applies here, except the full chromosome data from *Staphylococcus aureus LGA251* is used to build the statistical model, instead of only the intergenic regions.

Appendix C: 179 antibacterial candidate ORFs after filtering

Table 2: The 179 ORFs remaining after filtering. Acc. is accession number for the chromosome or plasmid sequence. Cont. is the container, i.e. Chromosome or Plasmid. pI is the peptide's theoretical isometric point. Avg. cons. is the average conservation of the ORF that coded for the peptide. Seq. is the peptide sequence. L. is peptide length. Upstream seq. + start codon is the ORF's 20 upstream nucleotides in addition to the ORF's start codon. The ORF DNA sequences are not shown.

Acc.	Cont.	pI	Avg. cons.	Seq.	L.	μH	OL	Upstr. seq. + start codon
NC_007622	Chromosome	10,53	957,04	MIFSQNLFRRPTPTCTLL	18	0,247	No	GACGTGGGAGTGGGACAGAAATG
NC_017349	Chromosome	10,79	900,16	LTRIEKKLVTSAFSFSQLLP I	21	0,374	No	GTCGTCCCACCCCGGCAAGGTTG
NC_017349	Chromosome	11,44	608,01	VQVGVGRRNKFYENIISVPL PRTTT	25	0,323	No	GAAAGTCAGCTTACAATAATGTG
NC_017349	Chromosome	9,26	561,39	MKVYPAQIREWDRNDIFAKF ISSSHPNLHIIVS	33	0,383	No	ACTAGCAATAAAGGGTTCAAATG
NC_017349	Chromosome	9,20	516,65	VLGPLTRIEKSLLQAHFRSV NY-CQYNFVEHRTLIYVPA	38	0,427	No	ATTTCTTTTCGAAATTCTCTGTG
NC_017349	Chromosome	10,80	502,57	VQVGGAPTQKLTKVSLQ	17	Too short	No	GAAAGACAGCTTACAATAATGTG
NC_020207	Chromosome	10,80	501,64	VFNQLTFGISSKNGKI	16	Too short	No	TCAAGACCTCTTGATACACGGTG
NC_017349	Chromosome	9,30	476,40	LRKYHFCPTPIGIYVV	16	Too short	No	GTGGGACGACGAAATAAATTTTG
NC_020207	Chromosome	12,28	316,46	LSHALNTYKRRKQKQLLRK	19	0,288	No	GACAAAAGTAAAGAACACTTTTG
NC_020207	Chromosome	9,15	309,50	MFPHIYIFPLIVKNSSSYFA VF	22	0,304	No	CTATTAAGGAAAAATCAACTATG
NC_017349	Chromosome	10,47	290,53	MCKLGWDNEINFAKISFLSH SHHKRRYIRHDEISSNLYKS CKRFTIR	47	0,437	No	CGAAAAGTCAGCTTACGATAATG
NC_017022	Chromosome	9,27	250,74	LFGCSYYLMQDSFFTTSFRL AL-NFLKK	27	0,385	No	AAAATTATGAGGAGCTATTTTTG
NC_020207	Chromosome	11,82	210,77	LNIIIKERGTKVKNTFVSRS KHV	23	0,298	No	ATTAGCCGTGTAAACAAATTTTG
NC_017337	Chromosome	10,42	210,26	MGPFLGCRLSLGLRLALLPQ ESRH	24	0,129	No	TATAAATAGAATTTTTGATGATG
NC_020532	Chromosome	10,48	203,02	MIDSKLWFFIFYGGKVKLPA FFNTQKAQLPL	31	0,257	No	TTTTTTTTATTAATTTAAAAATG
NC_017349	Chromosome	10,21	201,30	MSKREPKERKEASDCHKSRK VLSDDGSQLTFR	32	0,338	No	AACGAGTTTAGTAGAGCTAAATG
NC_017337	Chromosome	9,45	180,99	MYRTTSLTTCASWGGTTK	18	0,163	No	CATACTGATTGAAGACACTAATG
NC_022222	Chromosome	9,74	175,08	LYSVLNVYLTFKYFLVKVIL LM-RALD	26	0,305	No	ATGCGGAATAACGTGACATATTG
NC_020207	Chromosome	12,29	165,17	LSQSFFPNKWCSRSSFFGNK PKFHKNLRSNFRKFLLIS	38	0,475	No	AAAGGACTGTGACAAAAGTCTTG
NC_017022	Chromosome	10,46	165,12	VRQKCSLLLSHALFYR	16	Too short	No	CGTTTATACGTGTTTAGAGCGTG
NC_016942	Plasmid	10,78	163,03	LARLLREQDKLKTTG	15	Too short	No	ACATATAAATACAAAGTATTTTG
NC_017022	Chromosome	9,78	160,81	VNQTFISSVSNQRNHIPRKS LNYRTPIEIFLSYVQEAFYS SLI	43	0,361	No	AATCAATGGATTTTAGAGAAGTG
NC_021994	Chromosome	10,79	156,90	LVGRLTSLSGEKKGLFHAV	19	0,316	No	CATTAAAGAGAAAGTCACTCTTG

Continued on next page

Table 2 – continued from previous page

Acc.	Cont.	pI	Avg. cons.	Seq.	L.	μH	OL	Upstr. seq. + start codon
NC_020537	Chromosome	12,51	149,88	VKSTTRRTMSSPHHSSALTS TGFA	24	0,196	No	TAGAGGCTTTTCTCGGCAGTGTG
NC_017337	Chromosome	11,55	147,25	LRSQCGSSHPFFAWQRSTLA ERKLATIVAKNLS	33	0,255	No	AGACATAAAAAAAGAGACCTTG
NC_021059	Chromosome	10,55	136,66	LFSTFHGSSSFMVGR	15	Too short	No	AACAGAATATCAACTATTTATTG
NC_007622	Chromosome	10,55	130,92	LGSTPPTRIACRISFRNSLC WGP	23	0,121	No	GGGAATCCAATTTCTCTTTGTTG
NC_021994	Chromosome	11,01	126,95	LNLLKNSIQLTNQPVGKKID RLIKSKILWPSAYFASG	37	0,429	No	TAGATCAAGAAAATAAGCAATTG
NC_017333	Chromosome	9,79	124,21	MLNKHSKLNTICHVIPHLLK KMSFKTEYNMSRYSASSEED VPNISLERR	49	0,529	No	TTTAACCAAAAAATATTTGAATG
NC_022113	Chromosome	10,53	117,02	VTIACFFPLLRSRLLI	16	Too short	No	CTAAGAACCTTTCTTGACTTGTG
NC_017349	Chromosome	9,14	112,58	LIYVSGSYAVGAKIQL	16	Too short	No	ACTTTGTAGAGCATAGAACATTG
NC_017763	Chromosome	9,99	103,17	VGPNTENFEKKFYRQCKLGT IKKYFFFITLCLTHFPKY	38	0,422	No	CAGGCAATGCAAGTTGGGGTGTG
NC_017960	Chromosome	9,15	96,97	LYGFFHLHKIFYTLH	15	Too short	No	TACAGAAGGGACTTCCTTTTTTG
NC_019770	Chromosome	12,21	95,17	LFFNFWRISAYFRRTFNS	18	0,428	No	TAAGAAGTAATTTCCGAAAATTG
NC_021554	Chromosome	9,27	91,54	LGPQYMHFICYIKWFK	16	Too short	No	TCTTTTCGAAATTCTCTGTGTTG
NC_017022	Chromosome	9,17	84,31	VNWLSLTLFYRRDFLFCMDF FIYTKYFILSSKSDVHPKS	39	0,255	No	ATAGGGAAGTCCTCTTTTCTGTG
NC_021994	Chromosome	9,69	84,23	LSTLFKSFFDFVFRNLSATH LS	22	0,515	No	CATATCAGAGAGACAAGCGATTG
NC_017343	Chromosome	9,82	78,78	LKTIQRIIRGTCLWEVAFLG SPPNVVGI	28	0,38	No	TATTTATAGAAGCTACTTTCTTG
NC_020207	Chromosome	11,23	78,56	LHKFGLIPKNWLLNTVYQEV VSLPFSSESQGVGRRTLFIR KKGL	44	0,263	No	GAAAATGGCTTTCTACATTTTTG
NC_017960	Chromosome	11,53	78,37	VYRPPKVRFFGLTFGSRYIV TVLFSNKRCSPANSSVTSHN IQQT	44	0,205	No	GAGACTGTGACAAAAAACGTGTG
NC_017349	Chromosome	10,26	77,00	VWHEVCAISFLLCLRRVSIK KYFFFRN	27	0,223	No	TGAAGCGGTTCAAAAGAAGGGTG
NC_022222	Chromosome	9,42	74,32	LCDKEKIFLYRPQPAHYRKL TFRQLLCWGPHPNLHCL	37	0,198	No	TTTGGAAAGCGAGTGGGACATTG
NC_022443	Chromosome	9,13	74,13	LPKNLMSQPLFTLPKNLMSQ PLLLYLLELYISFINLLI	38	0,382	No	TAGAAATCAGCTTTTTTACATTG
NC_021994	Chromosome	9,93	73,03	LRKQPQKRGSIIYGTDESEL ILQPHFHFRY	30	0,275	No	ATGTACAAGATTTGCGCCATTTG
NC_022222	Chromosome	9,99	71,20	LKIILLLFFYEKKTIRYAYG SEKGSTIVTKKCISTC	36	0,225	No	AGCAGTAAGATATTTTCTAATTG
NC_020207	Chromosome	10,79	69,92	LSQPFLPSKRCSKASFSV	18	0,251	No	AAAAGAGGTTGTGACAATTTTTG

Continued on next page

Table 2 – continued from previous page

Acc.	Cont.	pI	Avg. cons.	Seq.	L.	μH	OL	Upstr. seq. + start codon
NC_018608	Chromosome	10,51	63,93	VTIACFFPLLRLSLTHLALL NSLRSFRQIQTFSLRQAIFL CVCFLF	46	0,345	No	CTAAGAACCTTTCTTGACTTGTG
NC_017960	Chromosome	10,64	62,40	LRTKFFYFAVCIIKRDK	17	Too short	No	AAAAAAACTTGGTACAGCGATTG
NC_016941	Chromosome	9,69	53,58	LTITNKILIIFVWFFT	16	Too short	No	TTAGAAAATGAGTTAATGAGTTG
NC_016941	Chromosome	9,73	50,81	MILCIYILNITNFTKIKNNS ILYYHLVFFNNTFVFLKSTN GGT	43	0,259	No	ATTCAGTTTTGTAAGTAAGAATG
NC_017349	Chromosome	10,42	411,08	MVANLRSARVERCQAS	16	Too short	No No	AAGAAAGGTTCTTAGCGACGATG, AAGAAAGGTTCTTAGCGACGATG
NC_022222	Chromosome	12,07	304,56	MLIRSINGETPEGAVPSRRP RL-RRHPRKAKPFNTKYCINR EQQ	43	0,305	No No	GTATTTCAAAGTAAAATTACATG, GTATTTCAAAGTAAAATTACATG
NC_022222	Chromosome	12,50	286,17	VPSQPRSSTGTAPSGVSPLI LRINM	25	0,241	No No	AATGGCTTCGCTTTCCTAGGGTG, AATGGCTTCGCTTTCCTAGGGTG
NC_017349	Chromosome	9,44	230,21	VKTFESDETRKERNEFSRAK	20	0,193	No No	AAGAAAAATGGCTTGGCGAAGTG, AAGAAAAATGGCTTGGCGAAGTG
NC_020532	Chromosome	10,04	243,74	LRFLCIKKSRKFYLPTIKDE EP	22	0,276	No No No	ACTTTAAATTATAGAGGCAATTG, ACTTTAAATTATAGAGGCAATTG, ACTTTAAATTATAGAGGCAATTG
NC_022222	Chromosome	9,30	110,05	MYKNYNMTQLTLPNRNFC	18	0,073	No No No	CTAAATTAACGAGGTGCCTTATG, CTAAATTAACGAGGTGCCTTATG, CTAAATTAACGAGGTGCCTTATG
NC_020207	Chromosome	10,34	112,12	VFESKPLKTEQSKNKLCSLR NIP	23	0,202	No No No No	CCAGTTTTCAATGAACAAAAGTG, CCAGTTTTCAATGAACAAAAGTG, CCAGTTTTCAATGAACAAAAGTG, CCAGTTTTCAATGAACAAAAGTG
NC_017022	Chromosome	10,84	211,21	LRVNLSKLNKVKTNCVVSVI FLRKEVIQPHLPIRLPCYQN	40	0,28	No No No No No	TTTTCAATGAACAAAAGTATTTG, TTTTCAATGAACAAAAGTATTTG, TTTTCAATGAACAAAAGTATTTG, TTTTCAATGAACAAAAGTATTTG, TTTTCAATGAACAAAAGTATTTG
NC_007622	Chromosome	10,64	869,26	LKKACYKRIFVQSTTANITS	20	0,107	No Yes No No	GTTGGGGCCCCTGACTAGAATTG, GTTGGGGCCCCTGACTAGAATTG, CCCGGCAAGGTTGACTAGAATTG, CCCGGCAAGGTTGACTAGAATTG
NC_017022	Chromosome	10,53	446,85	VGFTLRCFQRLSLPT	15	Too short	Yes	ATGGGAAATCTCATCTTGAGGTG
NC_021994	Chromosome	10,44	445,43	LAPRCRLVASWGCSRSQGLG CSPIKAARELVQNVVRQFGP YPSRALEI	48	0,542	Yes	GTCCACATCGACGGGGAGGTTTG

Continued on next page

Table 2 – continued from previous page

Acc.	Cont.	pI	Avg. cons.	Seq.	L.	μH	OL	Upstr. seq. + start codon
NC_021059	Chromosome	11,38	408,66	VIGHTGTETRSRLLREAAVG NLPQWAKA	28	0,263	Yes	TGCATAGCCGACCTGAGAGGGTG
NC_021059	Chromosome	11,65	407,80	LTGTRTSGGACGLIRSNAKN LTKS	24	0,364	Yes	AAGGTTGAAACTCAAAGGAATTG
NC_021059	Chromosome	11,25	400,77	LVRFFALLRIKPHAPPLVRV PVNSFEFQPCGRTPQAECLM R	41	0,348	Yes	TTGTCAAAGGATGTCAAGATTTG
NC_021059	Chromosome	12,98	399,22	VPAAAVIRRWQALSGIIGRK ARVGGFLSLM	30	0,465	Yes	AGAAAGCCACGGCTAACTACGTG
NC_021059	Chromosome	9,69	391,34	VFLHISAHFTATHGIPLSSS ALK-FSSFQ	28	0,212	Yes	GAAAGTCGCCTTCGCCACTGGTG
NC_020207	Chromosome	11,48	387,52	LGSPIRKSLDHSLRTAPQSI SVLVPSFIGS	30	0,439	Yes	GTAACATCCTATTAAAGATGTTG
NC_022222	Chromosome	12,25	289,12	MPLMIWATHVLPWTIQRAAK PRGQANPIKLFSVRIVVCNS TT	42	0,271	Yes	GGGGATGACGTCAAATCATCATG
NC_012121	Chromosome	11,62	269,87	MCSGEMRRDMEEHQWRRRLS GLQLTLMCETWGSNRIRYPG SPRRKR	46	0,338	Yes	GAAGAGGAAAGTGGAATTCCATG
NC_021670	Chromosome	12,25	263,82	VTGGAWLSSARVVRCWVKVP QRAQPLSLVAIIKLGTLS	38	0,417	Yes	CTTCCCCTTCGGGGGACAAAGTG
NC_018081	Chromosome	12,14	221,24	VAKAALWSVTDAEARKRGEQ NRIRYPGSPRRKR	33	0,145	Yes	AGATATATGGAGGAACACCAGTG
NC_020207	Chromosome	9,76	202,75	MGKAQTSKLACWGCRTPIW	19	0,091	Yes	CCTGAGTAGCGGCGAGCGAAATG
NC_013450	Chromosome	10,42	191,58	MRALRPIIPDNACHLRITAA AGT	23	0,219	Yes	TCAGACTTAAAAAACCGCCTATG
NC_017022	Chromosome	10,89	181,95	LLYGISTCFQVLSPSDGQVT HVLLTRSPLLFFRWSKLRWK KKRSTCMY	48	0,324	Yes	CAAAACCATGCGGTTTTCGATTG
NC_021994	Chromosome	12,37	157,64	LVTNGFPVRKFFSTVKRNRF KEVFREARKPLIKGSFNI	38	0,505	Yes	AAATATATATCAATATCATGTTG
NC_002952	Chromosome	10,92	156,56	LRDLTQHLTTRADDNHAPPV TLPPRRGRLYL	31	0,317	Yes	AAGCTTAAGGGTTGCGCTCGTTG
NC_002952	Chromosome	9,25	145,56	MWFNSKQREEPYQILTSFDN SRDRAFPFGGAK	32	0,39	Yes	ACCCGCACAAGCGGTGGAGCATG
NC_017960	Chromosome	11,48	143,16	MPLMTWATHVLQWEVQRVAK SRG	23	0,211	Yes	GGGGATGACGTCAAATCATCATG
NC_021994	Chromosome	11,48	126,47	LIRKALSGVTDGWTRGALAS W	21	0,44	Yes	ACAATCGAAACCGCATGGTTTTG
NC_017353	Chromosome	10,78	109,34	MLSGKDVALPRQLGCWLRSS HHLKSA	26	0,386	Yes	CAGCTAAGGTCCCAAAATATATG

Continued on next page

Table 2 – continued from previous page

Acc.	Cont.	pI	Avg. cons.	Seq.	L.	μH	OL	Upstr. seq. + start codon
NC_002951	Chromosome	11,01	105,18	VSKGVWKVVKYYRKHQRMLR NTIYYPAFNNGAIEGINNKI KLIK	44	0,507	Yes	ATTCAGTTAGTAAAAAGTCTGTG
NC_016941	Chromosome	10,14	102,86	MATKLKGCARCGTQPNISRH ELTTTMHHLSLCPPKGKALS LELSKDVKIW	50	0,252	Yes	TTAGAGTGCCCAACTTAATGATG
NC_022222	Chromosome	9,83	96,13	LLHQLRHHLHLYHLQLYLYL FVLSQQAFQRITLVFTAI	38	0,285	Yes	GAAATCATCATGATCCGTAATTG
NC_004461	Chromosome	12,33	95,87	LTARPTSRAGRKTDLVIRWF RMEGPSLNG	29	0,413	Yes	GTGTAAAGGCATAAGGGAGCTTG
NC_007350	Chromosome	11,84	89,81	LGCSPIKAVRELGSERRETV RSLSVVGVGNLRGAVLSTRG PG	42	0,455	Yes	GCTGTAGTCGGTCCCAAGGGTTG
NC_004461	Chromosome	10,95	64,65	VVWHLDVGSSHPGAVVGPKG WAVRPLKRYASWVQNVVRQF GPYPSWA	47	0,628	Yes	CAAGAGTTCACATCGACGGGGTG
NC_017349	Chromosome	9,66	293,37	MSTLIPKKRNYKLQTLFSIY ELIKHHNFYGEFDPGSG	37	0,234	Yes (partial)	ATTGAAAACTGAATGACAATATG
NC_017349	Chromosome	11,46	244,70	MTGSNRRPSACKADALPAEL IL-RFKTAWQRSTLAERKFDY HRR	43	0,362	Yes (partial)	GTAAGAATAAATGGTGGAGAATG
NC_017022	Chromosome	12,03	244,68	VRVGRRHAFRCYSGL	15	Too short	Yes (partial)	GTAGTGAGGGGTTGCCCCTTGTG
NC_022222	Chromosome	11,30	193,35	MAVSTGIEPAISCVTGRRVN RYTTRPIKYCGRRI	34	0,559	Yes (partial)	ACCATAATAAAGATGTAATGATG
NC_017341	Chromosome	11,58	184,07	LTYCHSVFNVHRVKNKWWRL AGSNR	25	0,39	Yes (partial)	ACGTTTTTTTGGAATTAACGTTG
NC_017022	Chromosome	10,12	166,78	VATSYSHKGQPLTTIGAKKL NFLCSAWLQVYPSRYRHHTV VLSFIE	46	0,373	Yes (partial)	TATGCCGGAATAACATCAGCGTG
NC_021994	Chromosome	10,45	165,66	LHAGGQRFDPARLHFI	16	Too short	Yes (partial)	CAGCTGGGAGAGCGCCTGCTTTG
NC_021994	Chromosome	9,46	135,70	LGPRGRRFESCLPDNYKFKH GALAQLGERLLCTQEVSGSI PLGSISFK	48	0,347	Yes (partial)	AGCTTGGTAGAGCACTTGGTTTG
NC_016941	Chromosome	9,50	91,80	LHAGGQRFDPASLHHLFTNY IRRCSSAG	28	0,483	Yes (partial)	CAGCTGGGAGAGCGCCTGCTTTG
NC_016941	Chromosome	10,86	90,92	MEGGRFELPNPKERIYSPPR LATSLPLHKNGAGQRT	36	0,333	Yes (partial)	TATGCCCCTATTAAAATTAAATG
NC_017349	Chromosome	9,83	89,43	VRDHRTGSIPVLGTILAPVA QL-DRAFDYGSRGYGFDSYRA RHF	43	0,454	Yes (partial)	ACAGGACTTAAAATCCTGCGGTG

Continued on next page

Table 2 – continued from previous page

Acc.	Cont.	pI	Avg. cons.	Seq.	L.	μH	OL	Upstr. seq. + start codon
NC_017349	Chromosome	9,34	84,06	LIRSQTLYPIELRALKWCRG PESNRYGDLSPQDFKSCASA SSATPAK	47	0,316	Yes (partial)	GGAGTCGAACCCATAACCTCTTG
NC_017349	Chromosome	10,92	84,02	VRLPVPPPRHYKNGAEDGIR TRDPNLGKVVFYR	33	0,327	Yes (partial)	ACCGCAGGATTTTAAGTCCTGTG
NC_020207	Chromosome	12,50	75,16	MVRVHLGPFILLARWSSG	18	0,28	Yes (partial)	GCCTGATAAGCGTGAGGTCGATG
NC_020207	Chromosome	11,22	73,78	VKGLEPPRRKALDPKSSASA NSATPAKI	28	0,248	Yes (partial)	GTAAAACTTAAATTATGCGGGTG
NC_021994	Chromosome	10,05	73,37	VFKFIIIGKTGFEPATPWSQ TKCSTKLSYFPFN	33	0,246	Yes (partial)	CCCAGCTGAGCTAAGGCCCCGTG
NC_016941	Chromosome	12,22	72,96	LNYAPIKIKWRGADSNCRTR RSGFTVRRV	29	0,207	Yes (partial)	TGGAGACCTCTATTCTACCGTTG
NC_020207	Chromosome	11,81	69,45	VITHRRILSPVRLPVPPPRR DCFGKAENGVRTRDPHLGKV VLYH	44	0,377	Yes (partial)	ACCGGAATCGAACCGGTACGGTG
NC_020995	Chromosome	9,83	60,74	MPNTEVKLLSADCSEGFPFV RVGRRHAFQFFRHSSVGSSA	40	0,297	Yes (partial)	AGAAGGATACACCTGTAACCATG
NC_020207	Chromosome	12,24	60,42	VVEHHLAKVGVAGSNPVFRF AKAIAPGWRNWQTHRT	36	0,565	Yes (partial)	AAACACGCGGAAATAGCTCAGTG
NC_022222	Chromosome	10,01	60,21	VGSIPTAPAMAAVVKWLTHR IVVPTFEGSIPFSRPYY	37	0,241	Yes (partial)	AGGTCTCCAAAACCTTTGGTGTG
NC_020207	Chromosome	13,35	59,11	LPRWGSRVRTPFSALPKQSR RGGGTGRRTGLKILR	35	0,523	Yes (partial)	CTCAGTGGTAGAGCACCACCTTG
NC_016941	Chromosome	10,89	57,99	VAPTHFPGVLLRPTRTTLLN NKYVIKKQKRYFTSVCMTPT GLEPVLPP	48	0,391	Yes (partial)	TTCGAACCCTCGAGACGCTTGTG
NC_021994	Chromosome	12,23	55,75	MVRVHLGPFNFIGPLVKRLR HRPFTAVTRVRIPYGSWKIL FEYLFLFI	48	0,46	Yes (partial)	GCCTGATAAGCGTGAGGTCGATG
NC_020207	Chromosome	11,48	53,33	LKLRVSAVRFCPAPPWRSSE VAKRDGL	27	0,26	Yes (partial)	TCAGTTGGTAGAGCAACGGATTG
NC_017960	Chromosome	12,23	534,87	LTGARTSGGACGLIRSNAKN LTRS	24	0,364	Yes Yes	AAGGTTGAAACTCAAAGGAATTG, AAGGTTGAAACTCAAAGGAATTG
NC_017960	Chromosome	9,50	531,98	MCSGEMRRYMEEHQWRRRLS GL	22	0,171	Yes Yes	GAAGAGGAGAGTGGAATTCCATG, GAAGAGGAGAGTGGAATTCCATG
NC_017960	Chromosome	9,69	529,31	VFLHISTHFTATHGIPLSSS ALK-SPSFQ	28	0,215	Yes Yes	GAGAGCCGCCTTCGCCACTGGTG, GAGAGCCGCCTTCGCCACTGGTG
NC_020207	Chromosome	11,15	486,57	MGSFRPVAGNLHLHRY	16	Too short	Yes Yes	TCAAACTACAGTAAAGCTCCATG, TCAAACTACAGTAAAGCTCCATG

Continued on next page

Table 2 – continued from previous page

Acc.	Cont.	pI	Avg. cons.	Seq.	L.	μH	OL	Upstr. seq. + start codon
NC_020207	Chromosome	11,38	484,66	VKMQVTRDRTERPHGALL	18	0,17	Yes Yes	TCGGTGAAATTTTAGTACCTGTG, TCGGTGAAATTTTAGTACCTGTG
NC_020207	Chromosome	9,84	481,58	MLSGKGCGVAQTTRMLA	17	Too short	Yes Yes	CAGCTAAGGTCCCAAAATATATG, CAGCTAAGGTCCCAAAATATATG
NC_017960	Chromosome	10,68	474,72	LKLKGIDGGPHKRWSMWFNS KQREEPYQVLTSFDHSRDRA SPSGAK	46	0,466	Yes Yes	TGGGGAGTACGACCGCAAGGTTG, TGGGGAGTACGACCGCAAGGTTG
NC_020207	Chromosome	10,46	466,82	LRGAVLSTRGPGWTYRWCTS CSAKGIAG	28	0,317	Yes Yes	CCGTCGCGGGCGTTGGAAATTTG, CCGTCGCGGGCGTTGGAAATTTG
NC_017960	Chromosome	11,96	457,73	MHSRPERVIGHIGTETRPKL LREAAVGNLRQWTKV	35	0,415	Yes Yes	AACGGCTCACCAAGGCCACGATG, AACGGCTCACCAAGGCCACGATG
NC_020207	Chromosome	9,53	424,13	VSERTLVKELGKMTP	15	Too short	Yes Yes	GTCGAGGAGAGAATCCTAAGGTG, GTCGAGGAGAGAATCCTAAGGTG
NC_020207	Chromosome	12,20	362,01	MATLTRTTNRGGRQCQMGSL TGAVAS	26	0,334	Yes Yes	TGGGATACTACCCCTGCGTTATG, TGGGATACTACCCCTGCGTTATG
NC_020207	Chromosome	11,30	325,79	LRQCPNRYAFRAGRNLPDKE FRYLRTVIVTAAVYWGFNSY LRLR	44	0,488	Yes Yes	AAATTTCACCGAGTCTCTCGTTG, AAATTTCACCGAGTCTCTCGTTG
NC_020207	Chromosome	9,34	288,58	LLVGVVGLQYGSSFR	15	Too short	Yes Yes	AAAAGCCCAAACCAGCAAGCTTG, AAAAGCCCAAACCAGCAAGCTTG
NC_007622	Chromosome	9,82	272,97	VPNTCKSSERTRSLLL	16	Too short	Yes Yes	TCAGGATGAACGCTGGCGGCGTG, TCAGGATGAACGCTGGCGGCGTG
NC_017349	Chromosome	9,31	268,52	VGYYPSCVGFLTRTTYRGGR QCQAGSLTGAVAS	33	0,378	Yes Yes	TAGCTTACGTGGAGGCGCTGGTG, TAGCTTACGTGGAGGCGCTGGTG
NC_017349	Chromosome	10,45	266,82	LSEFGNPRGAPRPNSALPPI IIT	23	0,221	Yes Yes	AATTAATTGGCATTCGGAGTTTG, AATTAATTGGCATTCGGAGTTTG
NC_020207	Chromosome	10,83	263,33	MWKTPRRILSTAEHEKFRRN PGGPSPKAKYSLVTDSEPVP	40	0,326	Yes Yes	GTAAAAACCCCGTAGACGAAATG, GTAAAAACCCCGTAGACGAAATG
NC_016941	Chromosome	10,31	112,66	MIARTCGALRSENAGVSSER RVRIPSTD	28	0,308	Yes Yes	CGTTCTAAGGGCGTTGAAGCATG, CGTTCTAAGGGCGTTGAAGCATG
NC_017022	Chromosome	10,76	104,36	LTWLKSISFGSPSFRRSPFE CSFNPRWEGVPGSAK	35	0,325	Yes Yes No	CGGATGAGATGAATCCTTGATTG, CGGATGAGATGAATCCTTGATTG, CGGATGAGATGAATCCTTGATTG
NC_017960	Chromosome	11,05	422,61	LGRVSVPMWPITLSGRLCIV ALVSRYLTN	29	0,185	Yes Yes Yes	CTGCTGCCTCCCGTAGGAGTTTG, CTGCTGCCTCCCGTAGGAGTTTG, CTGCTGCCTCCCGTAGGAGTTTG
NC_017349	Chromosome	9,59	302,63	LRGAVLSTRGPGWTYLWCTS CRANGIAG	28	0,276	Yes Yes Yes	CCGTCGTGGGCGTAGGAAATTTG, CCGTCGTGGGCGTAGGAAATTTG, CCGTCGTGGGCGTAGGAAATTTG

Continued on next page

Table 2 – continued from previous page

Acc.	Cont.	pI	Avg. cons.	Seq.	L.	μH	OL	Upstr. seq. + start codon
NC_017349	Chromosome	11,25	300,30	VLKASKHEAPLKMRFPNFGY KIPQR	25	0,235	Yes Yes Yes	GCTATGTGTGGACGGGATAAGTG, GCTATGTGTGGACGGGATAAGTG, GCTATGTGTGGACGGGATAAGTG
NC_017349	Chromosome	9,42	299,40	MLRCFQHLSRPHIATQLCRW HDNWYTRGMSIPVLSY	36	0,421	Yes Yes Yes	CTCATCTTGAGGGGGGCTTCATG, CTCATCTTGAGGGGGGCTTCATG, CTCATCTTGAGGGGGGCTTCATG
NC_017349	Chromosome	10,46	297,03	LSDGPSMRNHRITKSVFRPC STCRSRSQAPLCLYTL	36	0,432	Yes Yes Yes	CCCGGGGTAGCTTTTATCCGTTG, CCCGGGGTAGCTTTTATCCGTTG, CCCGGGGTAGCTTTTATCCGTTG
NC_017349	Chromosome	11,48	280,95	MMRLIGSRWKHGDMWS	16	Too short	Yes Yes Yes	GGTTATAAGATCCCTCAAAGATG, GGTTATAAGATCCCTCAAAGATG, GGTTATAAGATCCCTCAAAGATG
NC_017349	Chromosome	10,95	266,67	LSKGQSTASVICLAPVHFRR SVTRLVSYYALFK	33	0,342	Yes Yes Yes	CTTAGAACGCTCTCCTACCATTG, CTTAGAACGCTCTCCTACCATTG, CTTAGAACGCTCTCCTACCATTG
NC_017349	Chromosome	10,63	266,61	VKFRRNLGGPSPKAKYSLVT DSEPVP	26	0,266	Yes Yes Yes	TCCTGAGTACGACGGAGCACGTG, TCCTGAGTACGACGGAGCACGTG, TCCTGAGTACGACGGAGCACGTG
NC_017349	Chromosome	10,78	266,54	LVVDPKPGDLPLVRLKFR	18	0,064	Yes Yes Yes	CTGAATAGGGCGTTTAGTATTTG, CTGAATAGGGCGTTTAGTATTTG, CTGAATAGGGCGTTTAGTATTTG
NC_007622	Chromosome	11,36	264,70	VFLINSRLGLFTAALLGVNP KEHPFSRSYGVILPSSLTRV RSLTLEFSS	49	0,311	Yes Yes Yes	TTACGGTTTAGCAGAGACCTGTG, TTACGGTTTAGCAGAGACCTGTG, TTACGGTTTAGCAGAGACCTGTG
NC_017349	Chromosome	9,69	264,67	VMIIGGRALFGRGAPLGLPN SDKLRMPINLTWESEHG	37	0,173	Yes Yes Yes	AGCTTTAGGGCTAGCCTCAAGTG, AGCTTTAGGGCTAGCCTCAAGTG, AGCTTTAGGGCTAGCCTCAAGTG
NC_017349	Chromosome	9,50	254,37	VSYNPNKQACWFGLFPFRSP LLRESNFLSLPPGTKMFQFS GCAF	44	0,247	Yes Yes Yes	CCTTTGTAACTCCGTATAGAGTG, CCTTTGTAACTCCGTATAGAGTG, CCTTTGTAACTCCGTATAGAGTG
NC_017349	Chromosome	11,53	228,45	LLVGVVGHSIRSYKGRH	17	Too short	Yes Yes Yes	AAGAGCCCAAACCAACAAGCTTG, AAGAGCCCAAACCAACAAGCTTG, AAGAGCCCAAACCAACAAGCTTG
NC_020207	Chromosome	11,77	103,71	MATVFLTNFIRCRTSFRMLF RLSLTMSLLGCDDLRQELTV KVRLPR	46	0,38	Yes Yes Yes	TTTTATACGAAAAATCTGGAATG, CAGGTCTTTACTATTGTGGAATG, TTCATCACCATCAATCTGGAATG
NC_020207	Chromosome	10,42	69,40	MELCLFHFGSVQSRNSRLFL WRT	23	0,275	Yes Yes Yes	TATGTACGAGTCGGGAAAAAATG, TATGTACGAGTCGGGAAAAAATG, TATGTACGAGTCGGGAAAAAATG

Continued on next page

Table 2 – continued from previous page

Acc.	Cont.	pI	Avg. cons.	Seq.	L.	μH	OL	Upstr. seq. + start codon
NC_017022	Chromosome	9,14	116,95	MNIQTDSSEKMDYQKKWTST KSIKDSSHPLRPKETISLEN H	41	0,268	Yes Yes Yes No	GCCTTCCCAACGGGGATTAAATG, GCCTTCCCAACGGGGATTAAATG, GCCTTCCCAACGGGGATTAAATG, GCCTTCCCAACGGGGATTAAATG
NC_017960	Chromosome	10,45	534,38	VTGGAWLSSARVVRCWVKSR NERNPYC	27	0,33	Yes Yes Yes Yes	GCTTCCCCTTCGGGGGCAAAGTG, GCTTCCCCTTCGGGGGCAAAGTG, GCTTCCCCTTCGGGGGCAAAGTG, GCTTCCCCTTCGGGGGCAAAGTG
NC_017960	Chromosome	12,81	530,47	VPAAAVIRRWQALSGFIGRK ASAGGFLSLM	30	0,456	Yes Yes Yes Yes	AGAAAGCCACGGCTAACTACGTG, AGAAAGCCACGGCTAACTACGTG, AGAAAGCCACGGCTAACTACGTG, AGAAAGCCACGGCTAACTACGTG
NC_017960	Chromosome	9,81	506,23	VVKGCQDLVRFFALLRIKPH APPLVRAPVNSFEFQPCGRT PQAECLMR	48	0,361	Yes Yes Yes Yes	GGGGAAGCTCTATCTCTAGAGTG, GGGGAAGCTCTATCTCTAGAGTG, GGGGAAGCTCTATCTCTAGAGTG, GGGGAAGCTCTATCTCTAGAGTG
NC_020207	Chromosome	9,35	487,24	VLFTFPSRYWFTIGH	15	Too short	Yes Yes Yes Yes	TATTTCACTCCCCTTCCGGGGTG, TATTTCACTCCCCTTCCGGGGTG, TATTTCACTCCCCTTCCGGGGTG, TATTTCACTCCCCTTCCGGGGTG
NC_020207	Chromosome	9,67	483,89	VEKDVGLHRQLGCWLRSSHH LKSA	24	0,396	Yes Yes Yes Yes	GGTCCCAAAATATATGTTTAAGTG, GGTCCCAAAATATATGTTTAAGTG, GGTCCCAAAATATATGTTTAAGTG, GGTCCCAAAATATATGTTTAAGTG
NC_020207	Chromosome	9,85	459,99	VYSTASVICLAPVHFRRRVT RLVSYYALFKWWLLLSQHPS CLCNPTSFST	50	0,387	Yes Yes Yes Yes	TCCTACCAATACACCTAAAGGTG, TCCTACCAATACACCTAAAGGTG, TCCTACCAATACACCTAAAGGTG, TCCTACCAATACACCTAAAGGTG
NC_017960	Chromosome	10,24	455,75	LRSLRDLTQHLTTRADDNHA PPVTLPPKGKLYL	33	0,421	Yes Yes Yes Yes	ATGGCAACTAACAATAAGGGTTG, ATGGCAACTAACAATAAGGGTTG, ATGGCAACTAACAATAAGGGTTG, ATGGCAACTAACAATAAGGGTTG
NC_020207	Chromosome	9,67	390,67	VSDKIHSRKGNSPDHQLRSQ NIC	23	0,268	Yes Yes Yes Yes	TCATATCCGGGAGTCAGACTGTG, TCATATCCGGGAGTCAGACTGTG, TCATATCCGGGAGTCAGACTGTG, TCATATCCGGGAGTCAGACTGTG
NC_020207	Chromosome	9,66	221,11	MLLFLRTSCDGEGNNSTEVP DVTLPRKASSEKTAARTANR HR	42	0,304	Yes Yes Yes Yes	TAAGTCTGAAGAGGAGTCAAATG, TAAGTCTGAAGAGGAGTCAAATG, TAAGTCTGAAGAGGAGTCAAATG, TAAGTCTGAAGAGGAGTCAAATG

Continued on next page

Table 2 – continued from previous page

Acc.	Cont.	pI	Avg. cons.	Seq.	L.	μH	OL	Upstr. seq. + start codon
NC_020995	Chromosome	12,51	165,77	LSIAYACRPRLRSRLTLGGR AFPRKP	26	0,302	Yes Yes Yes Yes	GGTACAGGAATATCAACCTGTTG, GGTACAGGAATATCAACCTGTTG, GGTACAGGAATATCAACCTGTTG, GGTACAGGAATATCAACCTGTTG
NC_017022	Chromosome	9,50	491,43	LQAATRLHEAGIASNRGSAR RGEYVPGPCTHRPSHHESL	39	0,216	Yes Yes Yes Yes Yes	AAAGCTTCTCTCAGTTCGGATTG, AAAGCTTCTCTCAGTTCGGATTG, AAAGCTTCTCTCAGTTCGGATTG, AAAGCTTCTCTCAGTTCGGATTG, AAAGCTTCTCTCAGTTCGGATTG
NC_017022	Chromosome	10,02	239,23	VTNRRKVGMTSNHHAPYDLG YTRATMGSTTSCEVARLS	38	0,09	Yes Yes Yes Yes Yes	GCACTCTAGCAAGACTGCCGGTG, GCACTCTAGCAAGACTGCCGGTG, GCACTCTAGCAAGACTGCCGGTG, GCACTCTAGCAAGACTGCCGGTG, GCACTCTAGCAAGACTGCCGGTG
NC_017022	Chromosome	9,76	87,42	VPNTCKSNASFSTGACSTGK RGVANG	26	0,251	Yes Yes Yes Yes Yes	TCAGGACGAACGCTGGCGGCGTG, TCAGGACGAACGCTGGCGGCGTG, TCAGGACGAACGCTGGCGGCGTG, TCAGGACGAACGCTGGCGGCGTG, TCAGGACGAACGCTGGCGGCGTG
NC_007622	Chromosome	9,73	778,81	LTERKCACNKLFSILVNLAG VGRRNKFCENII	32			Nomatchesfound
NC_020207	Chromosome	10,79	292,45	MKRRDKQKEVVTEVVKKLSC IK	22	0,396		Nomatchesfound
NC_020207	Chromosome	9,22	224,89	MRNYFAVCFSNTQVKTLISQ PLFIYVLFCIPLFSLYYIK	39	0,223		Nomatchesfound
NC_017351	Chromosome	9,06	216,95	MYHFNITKNRYPFITYENAL VTSFFQF	27	0,216		Nomatchesfound
NC_020532	Chromosome	10,47	198,36	MKSFIKVLHQMWWVYNLKNY FKLQLLEFLLFKL	33	0,306		Nomatchesfound
NC_020207	Chromosome	10,51	197,20	LTKSCPASSNKNSQAKIAPH IFRLVGLITADASF	34	0,386		Nomatchesfound
NC_017960	Chromosome	10,49	188,74	MKKNTLLTCDLCSFFIFYKK RGCEKAALHQVIRTAKQK	38	0,247		Nomatchesfound
NC_020207	Chromosome	9,90	187,10	VICVLFLFFMCYSFSNSKAK GDVIKKEVVKKLPCIK	36	0,239		Nomatchesfound
NC_017349	Chromosome	10,29	183,47	LRKYHFCPTPIKYMRSNKHY PQLSNLKLSLTETKVNEVLD R	41	0,276		Nomatchesfound
NC_021994	Chromosome	9,41	173,02	MVHSNCISDERKQRSVTITG KPLVTNMILIYILRTLEHLL NAH	43	0,394		Nomatchesfound

Continued on next page

Table 2 – continued from previous page

Acc.	Cont.	pI	Avg. cons.	Seq.	L.	μH	OL	Upstr. seq. + start codon
NC_022604	Chromosome	11,12	169,71	VTKPKLKETSAPISFGYRNF NNFKARIMMIFSLYKGEKKK TTKPNNGLAA	50	0,431		Nomatchesfound
NC_010079	Chromosome	9,53	95,94	MDWPPNNKSSKRGFEPLTL	19	0,237		Nomatchesfound
NC_016928	Chromosome	9,22	91,98	LAWDIKFLGNVKKLISINYL IENGLPSFS	29	0,379		Nomatchesfound
NC_021059	Chromosome	10,25	84,47	LLKRYLIANSMNSDIIVAQH PIIQKFNW	28	0,288		Nomatchesfound
NC_021994	Chromosome	10,35	81,73	MIKSPQNKQLRGRYAILVEA FRDKDPDYFSPYWQNFPKR	39	0,42		Nomatchesfound
NC_016941	Chromosome	10,25	71,55	LKIILLLFFRDLCPSLMTSI NKSHSFYRFPL	31	0,485		Nomatchesfound
NC_018221	Chromosome	9,14	61,61	VKAYGYTPLFSFVGNTI	17	Too short		Nomatchesfound
NC_017022	Chromosome	9,68	61,34	LVVVNFILQKGLPFCMDFFI LHKIFSHQLFLRILLSMHSP TSWG	44	0,265		Nomatchesfound
NC_004461	Chromosome	11,96	58,68	LEIAGSLRNSFRASLNKVRV RKGNSPDHQLRSQNIC	36	0,493		Nomatchesfound
NC_017347	Chromosome	9,18	52,68	MPDLIEMIVFKVFTSWRGPN TEADRKSAYNNVQENFKRNS TDNASWGSTK	50	0,405		Nomatchesfound

Norges miljø- og
biovitenskapelige
universitet

Postboks 5003
NO-1432 Ås
67 23 00 00
www.nmbu.no