UNIVERSITY OF NOVA GORICA
GRADUATE SCHOOL

COMPUTATIONAL ANALYSIS OF QUORUM SENSING SYSTEMS IN BACTERIAL GENOMES: DEVELOPING AUTOMATED ANNOTATION TOOLS

DISSERTATION

Sanjarbek Hudaiberdiev

Mentor: Prof. Sándor Pongor

Nova Gorica, 2014

To my mother and father.

Апама жана атама

Abstract

Genome sequencing technologies produce large amounts of genomic data that allow biologists to study novel problems, such as microbial communities that consist of thousands of species communicating with each other using various signals. This problem is important in medicine, agriculture and the environmental sciences. Currently, new bacterial genomes are being sequenced by the hundreds but genome annotation – predicting the location and function of the genes – is a bottleneck. My thesis project deals with the annotation of genes involved in bacterial communication, in all bacterial genomes available in the databases. This is a long term project, my work concentrated on the study of quorum sensing (QS) genes in proteobacteria that help bacteria to stage a density dependent response, based on a well-defined family of chemical signals, *N*-acyl homoserine lactones (AHL). My goal was to develop an automated system that allows one to find QS genes of AHL systems in complete and draft genomes – QS genes are poorly annotated in the current databases. This is a so-called subsystem based approach because we do not annotate one entire genome, but a subsystem in many genomes.

In my thesis project I constructed an automated pipeline for analyzing complete and draft genomes, which was primarily based on the known technique of Hidden Markov Models (HMMs). I contributed original software tools for the analysis that allow one to study the topology of QS systems within the chromosome, to classify and visualize the local topology of QS genes and to study horizontal gene transfer in these systems. I used the tools to analyze all currently available complete genomes (2620) and draft genomes (6970). In particular, I analyzed a family of solo *LuxR* proteins, a specific group of QS genes that occur „alone" in the chromosome, not in close proximity of other QS genes. I found that these genes cluster into individual subgroups, and sometimes form a novel local arrangement I called "twin *luxR* topology". I studied the sequence variability of these novel protein families and made the results available via a common web server. My results were published in 4 publications and there is one more in preparation.

Izvleček

Tehnologije določanja DNA zaporedij omogočajo generacijo obsežnih količin informacij o genomu, ki biologom omogočajo raziskave problemov, kot so na primer raziskave mikrobnih združb. Mikrobne združbe sestojijo iz predstavnikov več tisoč različnih mikrobnih vrst, ki za medsebojno komunikacijo uporabljajo različne signale. Komunikacija med predstavniki mikrobnih združb predstavlja problem predvsem v medicinskih, kmetijskih in okolijskih znanostih. Raziskovalci trenutno določajo DNA zaporedje velikemu številu mikrobnih genomov, anotacija genov pa še vedno predstavlja ozko grlo analize. V doktorskem raziskovalnem delu se osredotočam na anotacijo genov odgovornih za komunikacijo med bakterijskimi vrstami, ne glede na to ali je genom teh vrst že anotiran ali šele v »draft« stanju. Anotacija vseh genov je dolgotrajen projekt, zato sem se pri svojem delu osredotočil predvsem na gene, ki kodirajo proteobakterijske N-acil homoserin-laktone (AHL). AHL sodelujejo v procesih zaznavanja celične gostote (quorum sensing, QS) in omogočajo skupinski odziv ob primerni gostoti celic. Anotacija QS genov v obstoječih podatkovnih bazah je skopa. Cilj mojega raziskovalnega dela je bil zato razvoj avtomatiziranega sistema, ki bi omogočal identifikacijo AHL genov v katerem koli mikroorganizmu, ne glede na stopnjo anotacije genoma tega mikroorganizma. Ker se osredotočam na anotacijo podsistema genov med številnimi različnimi organizmi, ne pa na anotacijo vseh genov v določenem organizmu, ta pristop predstavlja tako imenovan podsistemski pristop.

Razvil sem platformo za avtomatizirano obdelavo anotiranih in »draft« genomov, ki temelji na prikritih Markovih modelih (HMM). Platforma vsebuje programska orodja, ki omogočajo raziskave topologije QS sistemov znotraj kromosomov, klasifikacijo in vizualizacijo lokalne topologije QS genov ter raziskave horizontalnega prenosa genov znotraj mikrobnih združb. Razvita programska orodja sem uporabil za analizo 2620 anotiranih in 6970 »draft« mikrobnih genomov, ki so trenutno dostopni v podatkovnih bazah. Osredotočil sem se na analizo solo luxR proteinov, ki predstavljajo specifično skupino QS genov. Na kromosomu se nahajajo posamično, oddaljeni od drugih QS genov. Odkril sem, da ti geni tvorijo različne podskupine in občasno tvorijo strukture, ki sem jih poimenoval »twin luxR topology«. Raziskoval sem variabilnost DNA zaporedij teh novih proteinskih družin in rezultate objavil na spletnem strežniku. Raziskovalno delo sem objavil tudi v sklopu štirih že objavljenih člankov, za objavo pa pripravljam še en članek.

List Of Publications

Some parts of this thesis were published as or partially contributed to the following publications:

1. Hudaiberdiev, S., Choudhary, K.S., Vera, R., Gelencser, Z., Lamba, D. and Pongor, S. (2014). Census of solo LuxR genes in becteria. (in progress)
2. Choudhary, K.S.; Dogsa, I.; Marsetic, Z.; Hudaiberdiev, S.; Vera, R.; Pongor, S.; Mandic-Mulec, I. (2014). ComQXPA quorum sensing systems may not be unique to Bacillus subtilis: A census in prokaryotic genomes. *PlosOne, 9(5), e96122.*
3. Choudhary, K.S., Hudaiberdiev, S., Gelencsér, Z., Coutinho, B.G., Venturi, V., Pongor, S. (2013). The organization of the quorum sensing luxI/R family genes in *Burkholderia. Int. J. Mol. Sci.*, *14*(7), 13727-13747.
4. Gelencsér, Z.; Choudhary, K.S.; Coutinho, B.G.; Hudaiberdiev, S.; Galbáts, B.; Venturi, V.; Pongor., S. (2012). Classifying the topology of AHL-driven quorum sensing circuits in proteobacterial genomes. *Sensors. 12(5), 5432-5444.*
5. Gelencsér, Z.; Galbáts, B.; Gonzalez, J.F.; Choudhary, K.S.; Hudaiberdiev, S.; Venturi, V.; Pongor, S. (2012). Chromosomal arrangement of AHL driven quorum sensing circuits in *Pseudomonas. ISRN Microbiology. 2012, 6.*

List of Abbreviations

ORF	Open Reading Frame
MM	Markov Model
HMM	Hidden Markov Model
QS	Quorum Sensing
AHL	*Acyl-homoserine lactone*
R,I,M,L	*luxR*,*luxI*,*rsaM* and *rsaL*
DSF	Diffusible Signaling Factor
BDSF	*Burkholderia* Diffusible Signaling Factor
PQS	*Pseudomonas* Quinolone Signal
HGT	Horizontal Gene Transfer
KL	Kullback-Leibler divergence
LDA	Linear Discriminant Analysis

List of figures

List of tables

Acknowlegements

Contributions and supports of many people made accomplishment of this study and writing of this thesis possible. First of all, I would like to thank my supervisor Prof. Sándor Pongor for his supervision, guidance and patience throughout my study. His particular way of approaching scientific problems and searching for answers always inspired me to be a better researcher and to be more articulate in formulating questions and answers to them. I express my gratitude to Dr. Vittorio Venturi, for his mentoring and valuable career recommendations.

I would like to thank my fellows from the group, Attila, Sonal and Roberto for their support and company. The family atmosphere that we succeeded to create will always be in my memories.

All of my friends, flat mates and colleagues in Trieste and ICGEB, who shared their time with me during these three years made my social life joyful and colorful. Thank you guys all!

Finally, I would like to thank ICGEB, University of Nova Gorica and Italy for giving me opportunity to spend three amazing years in Trieste, combining my studies with stunning experiences.

1. Introduction

"Begin at the beginning," the King said, very gravely, " and go on until you come to the end: then stop" – Lewis Carroll, *Alice in Wonderland.*

1.1. Sequencing of DNA

Not even 20 years passed since the complete genome of the bacterium *Haemophilus influenzae* [1] was sequenced, which was the first event of its kind, and now we have more than 2500 complete bacterial genomes. In addition we have over 150 complete eukaryotic genomes. And the number of ongoing sequencing projects is above 15000 [2] . In order to arrive to this point, several important milestones were achieved, but the most important of them was doubtlessly the event in June 2000, when the president of United States and British Prime minister have jointly announced the end of first phase of Human Genome sequencing project, thus revealing human genome. This was only very raw draft format, but in the subsequent years, the results were organized into a full genomic form.

With the fast development of sequencing methods, the number of published sequences is growing at exponential pace. While the earlier sequencing efforts almost exclusively used so-called *chain termination* sequencing method developed by Sanger et.al [3], today most sequencing projects use so-called next generation sequencing techniques [4], the biggest advantage of these methods is that they are in order of magnitude less expensive than the previous ones. While in 2001 sequencing of raw mega base-pairs cost around 5300$, in 2012 this service could be reached at a cost of 0.09$[5], (Figure 1). The decrease in sequencing cost led to a fast increase of known DNA sequences. The GenBank database of the NCBI (National Canter of Biotechnology and Information, USA) is one of the most important primary sources of DNA sequences [6] and the size of this database has reached by 2008 the size of 100 billion base pairs, while 10 years before that it was only 1-2 billion base-pairs (Figure 2).

The fast increase in amount of sequences was only one side of the coin. We also have to understand the functions of the genes: it makes no sense to have only the full sequence of human genome if we don't understand the structure and the function of the genome, since otherwise it will remain as a mere sequence of characters without any biological information. The tool of understanding genome sequences is *genome annotation*.

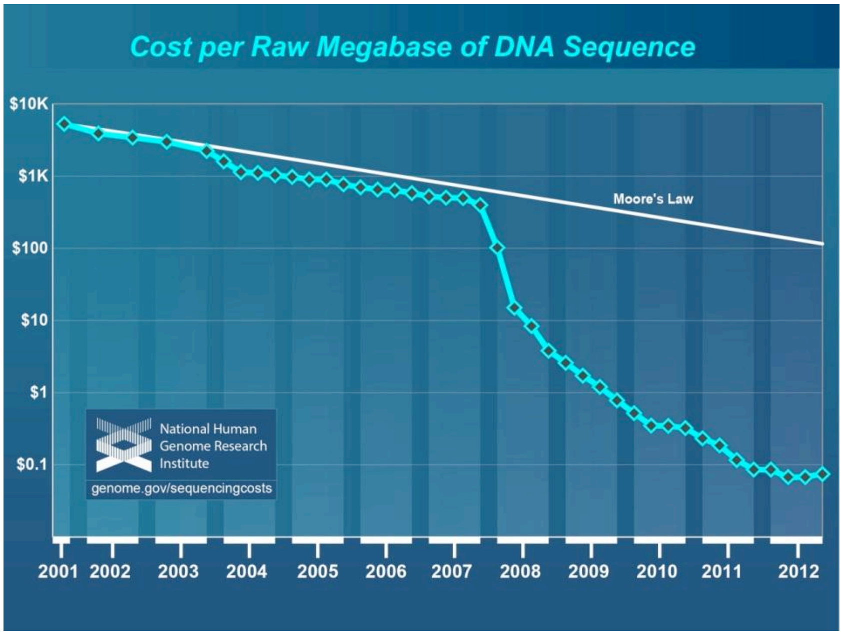


Figure 1. The cost of sequencing [7]. The price of sequencing one mega-base of DNA, in comparison with Moore's Law.

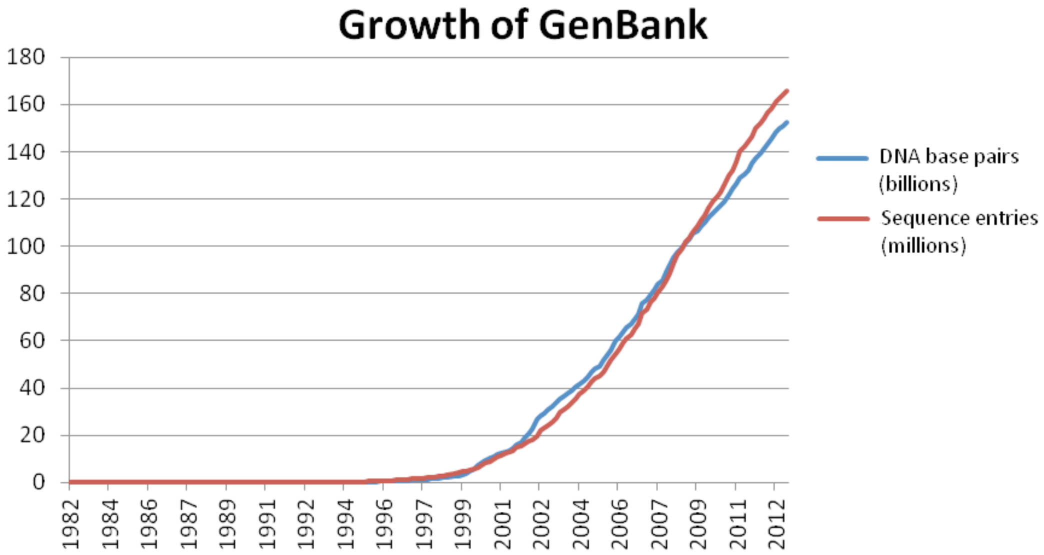


Figure 2. The growth of GenBank [8]

1.2. The concept of genome annotation.

Informally the term *genome annotation* denotes a process whereby we assign additional pieces of information to a raw genomic sequence. The definition of www.medicinenet.com is a typical example of such an informal definition: ”Genome annotation: the process of identifying the locations of genes of all of the coding regions in a genome and determining what these genes do. Annotation (irrespective of the context) is a note added by way of explanation or commentary. Once a genome is sequenced, it needs to be annotated to make sense of it”. Even though some annotation is added to the raw data already during the process of sequencing, more serious annotation work starts only when the raw data is added to a database. This is especially true when the data is deposited to a public database accessible to everyone. The initial information added to raw data includes the host organism, the number of sequencing experiment, the time of the sequencing which means the(Altschul, Gish et al. 1990) general bookkeeping data. During genome annotation, which is carried out for including this data to a database, we add a number of different kinds of information and these pieces of information are added by professional annotators who have deep biological knowledge. Annotators use various electronic databases and bioinformatics tools and the results they find are then described in terms of well-defined vocabulary which is contained in ontologies or other semantic resources.

The formal definition of genome annotation cannot be found in the literature so here I will be using conceptual framework which is based on the bioinformatics courses held at ICGEB, Trieste.

In bioinformatics, the concept of structure can be regarded as an ensemble of entities (substructures) and relationships, where the latter denotes the connections between the entities. The description of such structure can be simplified, for instance a structure can be described only in terms of its composition which includes the number of constituent units without the relationships between them. In addition there are even simpler property-like descriptions, for instance the sequence is helical or hydrophobic etc. The allowed names of the entities and the allowed properties as well as relations that are allowed between entities are included in bioinformatics ontologies. The description of such a structure can be formally regarded as an entity-attribute value data item [9] and the same kind of descriptions belong also to the relations. This general description is important because it can be adapted to all important data types of bioinformatics, which include sequences (DNA, RNA or protein), 3D structures, networks as well as textual items. For instance genomes can be represented in terms of sequences and in a sequence the units are nucleotides (A,C,T,G), and from the relations the only one type which is included denotes the vicinity within the chain – and this relationship is not even separately represented in a sequence. Nucleotides are defined in terms of a well-defined alphabet – for instance the IUPAC nomenclature – and as a result the sequence is represented as a series of characters.1.3.

In the context of genome annotation the term *raw data* refers to a series of characters. The theoretical topology of such a data structure is the line of numbers which can be best pictured as a series of empty positions, and in the course of genome sequencing we assign nucleotide character to these empty positions. The genome may consist of one single sequence which can be linear or circular, but it also may consist of several sequences. In the bacterial world, a typical bacterial genome consists of one single linear or circular DNA sequence, which is the genome, and in addition it can contain one or more circular plasmid sequences as well. Some of the more complicated bacterial genomes like the one of the *Burkholderia* genus consist of several large chromosomes and several plasmids. Once the genome is sequenced it can be either complete, which means that it is assembled into a full chromosome, or a draft genome which consists of several, sometimes overlapping, so-called contig sequences. Both the complete genomes and the draft genomes can be annotated or left in their raw forms. Finally, there are individual gene sequences which are deposited in databases such as RefSeq or GeneBank and which are result of individual small scale sequencing efforts where the target is a small segment of a bacterial chromosome.

In the course of genome annotation we assign attributes – or as we call them in bioinformatics: *descriptors* – to a raw sequence or to a given part of it. Descriptors can be categorized as global descriptors, which refer to entire genome, and local descriptors, which only refer to part of them (Figure 3)

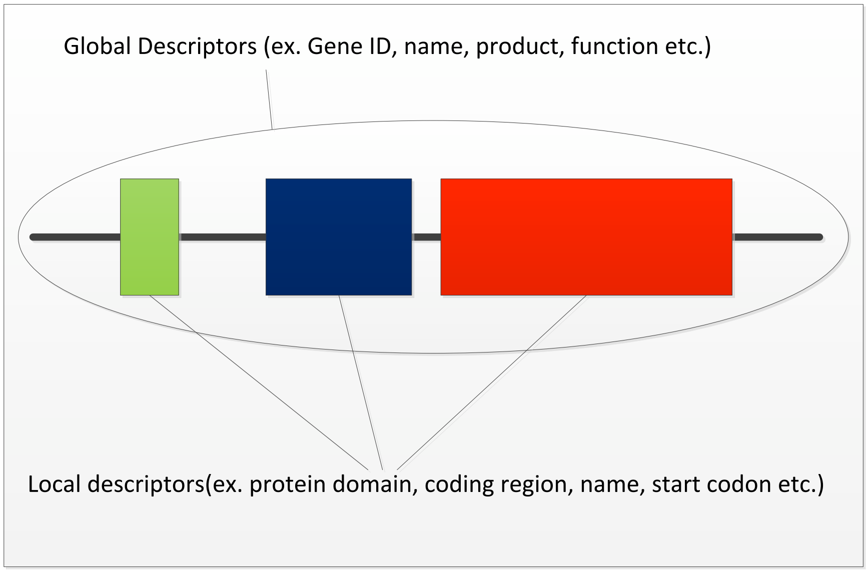

Global Descriptors (ex. Gene ID, name, product, function etc.)
Local descriptors(ex. protein domain, coding region, name, start codon etc.)

Figure 3. Classification of descriptors.

The sources of descriptors are the following:

1. *Human knowledge*. This is formulated in terms of a fixed vocabulary given as ontologies and the form of such a description can be a free text. But more often annotators use a given set of free structures.

2. *Computational procedures*. There are two kinds of such procedures; either we use a database to assign some kind of a description based on similarity (i.e. putative protease) or we can carry out calculations on the sequence, for instance we determine a low complexity region. While the first type includes databases and similarity search programs the second is based only on the sequence or on the structure.

3. *Database cross references*. When we add cross references we connect a piece of raw data or a part of it to a data item in another database using a pointer. The descriptors in this second database are from the same data sources mentioned above.

On the basis of forgoing we can make a logical sketch of DNA sequence annotation. A DNA sequence can be considered as annotated if the coding regions and other gene sequences are marked in it along with other segments of known structure or function. In addition the coding genes must be cross linked to as many databases as possible:

1. To primary databases such as protein sequence database UniProt.
2. To secondary databases such as: a) sequence databases clustered according to their functions, such as the protein sequence database COG, b) sequence databases clustered according to their structures such as PFAM or SBASE.

This sketch has two important consequences: 1) An annotation is never “complete” since many new proteins have no records in the primary or secondary databases or may have a corresponding un-annotated record. 2) Annotations change very frequently because the background databases to which an annotated item is cross linked are continuously updated or even changed. For this reason it's always very hard to claim that a genome is completely annotated (Figure 4).

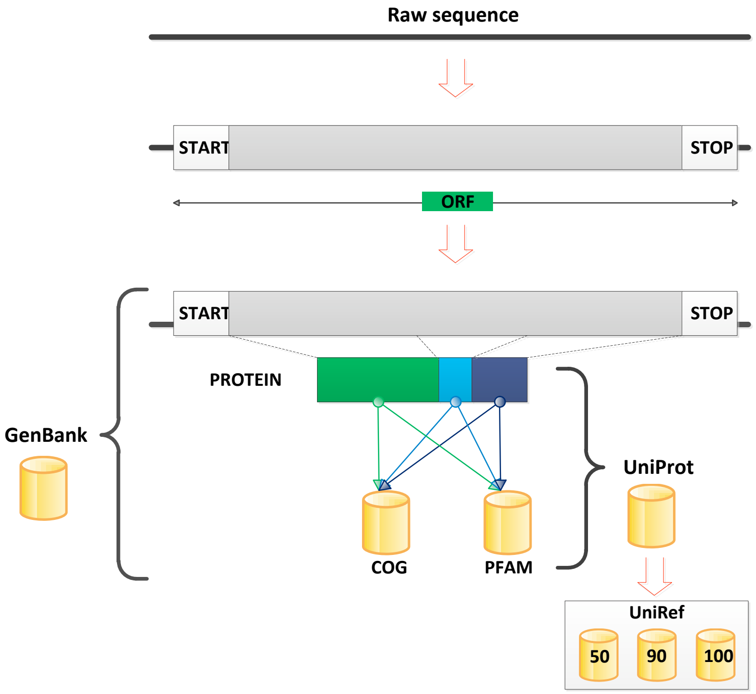


Figure 4. Steps of genome annotation. The steps of genome annotation are shown here as projected to a few commonly used bioinformatics databases. In a raw genome sequence we first find the positions of potential genes i.e. so-called Open Reading Frames (ORF). After that, we look for substructures of a protein coding gene ex. protein domains. And cross-link these domains to know secondary databases such as COG, PFAM. After that, an annotated protein sequence is inserted into the UniProt [10] database, as well as UniRef [11] which contains pre-clustered forms of protein sequences.

The idealized situation depicted in Figure 4 can be reached only with well-organized and adequately updated integrated databases. However, one can observe that using currently available tools and above mentioned approach, not always it is possible to reach the maximum coverage of annotation. In practice, an annotated genome is such a genome sequence which is stored in the complete genomes section of sequence databases. The best collection of such genome sequences is the NCBI's collection of complete genomes. And in this collection we have two categories for bacterial genome sequences:

1. *Complete genomes*. In these genomes, the positions of all the genes are determined and the part of them, usually more than half of them, are provided with some functional descriptions. The sequence of such complete genome is validated. So it is devoid of obvious sequencing errors.

2. *Draft genomes*. These consist of several unassembled sequence segments, so-called *contigs.* Some of the contigs are annotated, i.e. have an annotation file (so-called *ptt* file) which is described below.

In addition, we naturally have DNA sequence records of the old-fashioned sequence collections such as GenBank. The annotational part of a GenBank sequence contains the positions of the protein or RNA genes. As well as an accession number which allows one to link the GenBank record and a particular protein sequence to a sequence database like UniProt. On the basis of above definitions we can formulate the process of genome annotation from the point of view of data formats. Raw data formats are deposited in the simplest data formats such as *fasta* files or concatenated fasta files. In these files sequence contains one single annotation line and this can contain the number of the experiment, or in the case of better annotated sequences, some gene identifiers that link the sequence to other databases. When such a DNA record is deposited into a public database, ex. NCBI, it gets an unchangeable identifier which will appear in the annotation line of the fasta file.

Full genomes are deposited in the following data formats:

faa: This is a concatenated fasta file which contains the protein sequences belonging to bacterial chromosome or plasmid. Each sequence has a separate annotation line that contains more important IDs of the sequence and its potential functions.

ffn: This a concatenated file similar to the previous one but it contains nucleic acid sequences of the genes, without further pieces of information.

fna: This is a continuous DNA sequence without annotations or segmentations.

The annotation line contains only the name of the bacterium and the stage of sequencing i.e. if it's completed or not.

gbk: This is the GenBank record of a given bacterial chromosome or plasmid, which contains all annotation information that refers to the subject sequence including the name of that bacterium, full taxonomy, identifiers, journal references, the positions of reading frames as well as their functional annotations. This format is more transparent for a human reader but it's more complex to parse computationally comparing to previous formats.

ptt: This is a table-like format which is similar to feature table of SwissProt database [12] and this table contains the genes and other similar features of DNA in serial order of their sequence positions. Each line corresponds to a gene and includes information about that gene such as from/to coordinates, identifiers, potential function, names, references to COG, PFAM and UniProt databases. This format is ideal for information retrieval because its format is handy for parsing computationally and contains little unnecessary information.

The annotation level of draft genomes is between the raw data and complete genome, which means that for some contigs we find all the above files, whereas for some we do not have anything.

Bacterial genomes are 5 million base-pair long typically and contain 3-5 thousand genes. A well annotated bacterial genome, such as those which are included in NCBI's database, contain typically 25% and sometimes 40% un-annotated genes, and these are tagged as “gene of unknown function”, ”hypothetical protein” etc. In general, the core genes of bacterial genomes are annotated in details which are present in most bacterial species and carry out typical core cellular functions. Sometimes these genes are referred to as *housekeeping* genes. The *accessory* genes, which are also referred to as the *shell,* carry out special functions of a bacterial cell which can sometimes be found only in one species. The shell genome is usually much less annotated than the core genome. One reason for this situation is that although the shell genes are computationally predicted very well, they don't provide the same level of confidence, due to their species-specific nature. Thus, they are not always annotated by annotators as often as the core genes are.

As for the workflow of the genome annotations, we have two main approaches. One is the full annotation of a complete genome, which involves annotating as many genes as possible throughout the genome and leaving as few genes un-annotated as possible. The second approach refers to annotating subsystems, ex. a metabolic pathway, but in all possible genomes. In bacterial genome annotation this approach is especially important given the fact that a large number of un-annotated genes found in their genomes.

In case of full genome annotation, we choose an un-annotated genome and try to determine the functions of as many genes as possible. We do it using databases and several searching algorithms. In this case, we need to rely on the content of the genetic databases and if the content of the database is incorrect, we will necessarily commit a mistake, which is a problem of similarity based genome annotation in general. If we carry out similarity searching, the usual result is the nearest neighbor of a given gene. The fact that we found a nearest neighbor doesn't guarantee that the function of the new gene will be the same of its neighbor, so we have to determine using other means if the function is transferable. More often the function is automatically transferred if there is a more than 90% or more sequence similarity or in terms of blast e-values smaller than 10^{-4}. But these automated function assignments can also lead to false positive or false negative assignments, which is a general problem. And moreover, the usage of such automated function assignment tools is the reason why such large portions of bacterial genes are left un-annotated.

On the other hand, if we annotate subsystems, we do not work with single genome, but on a given subsystem such as a biological pathway or a signaling mechanism etc. [13]. When we already know some genes which belong to subsystem that we want to annotate, we will try to locate similar genes in the genome and we will normally also check if these genes are in the same local arrangement as the genes in already known genomes. In other words, the validation of the gene function relies not only on similarity, but also on similar arrangement within the genomes, which helps one to filter out false positive and false negative results which would necessarily be present were the process be based simply or solely based on sequence similarity. Nevertheless, this method can also fail because we can also find new hitherto unknown variations of a subsystem or if the subsystem is simple, for instance if it consists of only two genes, spurious similarities can also lead to errors.

1.3. Tools of genome annotation.

1.3.1. Pairwise sequence alignment.

Similarity search techniques between two sequences constitute one of the most important cornerstones of bioinformatics. On the computational level, the process of similarity searching is done by modified string matching algorithms, called pairwise sequence alignment algorithms (referred to as plain sequence alignment further).

Sequence alignments can be in two types: global and local alignments, and can be used for aligning DNA, RNA or protein sequences. While both types of alignments are similar to each other on algorithmic level, and are both called *aligning algorithms*, they answer fundamentally different questions. The global alignment score tells us how similar the given sequences are, while the local alignment not only gives the similarity but also helps us to investigate, if the given sequences share conserved regions or not. These two methods work as follows:

Global alignment: When performing global sequence alignment, algorithm reads the given two sequences and starts aligning them by matching the first residues of the sequences. Further it keeps scanning iteratively over all the residues by either matching two residues or shifting the sequences by leaving that point either unaligned or aligning one of the residues with an empty gap. So, in this way, global alignment *forces* the input sequences to cover each other to maximum extend, making them aligned globally. If we assume that aligned sequences evolved from a common ancestor, the mismatches are mutations, and gaps are indels (insertions and deletions). For calculating the overall score of the alignment, so-called substitution matrices are used. The most frequently used substitution matrices are block substitution matrix (BLOSUM) [14] and point accepted mutation (PAM) [15] matrix series. While identical matches and mismatches are scored using the corresponding values from substitution matrices, the indels are penalized (using negative scores). And the penalty for gap opening is higher than extending an already existing gap. The resultant score will be the sum of the scores corresponding for each position on the alignment. This approach for aligning is most often carried out by algorithm called Needleman-Wunsch which is based on dynamic programming [16].

Local alignment: In this type of alignment, algorithm does not try to stretch the alignment to full coverage, instead searching for most similar subsequences between two input sequences. On the algorithmic level, this effect is achieved by penalizing the gaps with smaller penalties comparing to global alignment. When calculating the resultant similarity score, it sums up the scores of aligned blocks, which are calculated using the same mechanism as global alignment. The most prominent implementation of this approach is Smith-Waterman algorithm which is also based on dynamic programming [17].

The difference between global and local alignments is depicted in Figure 5.

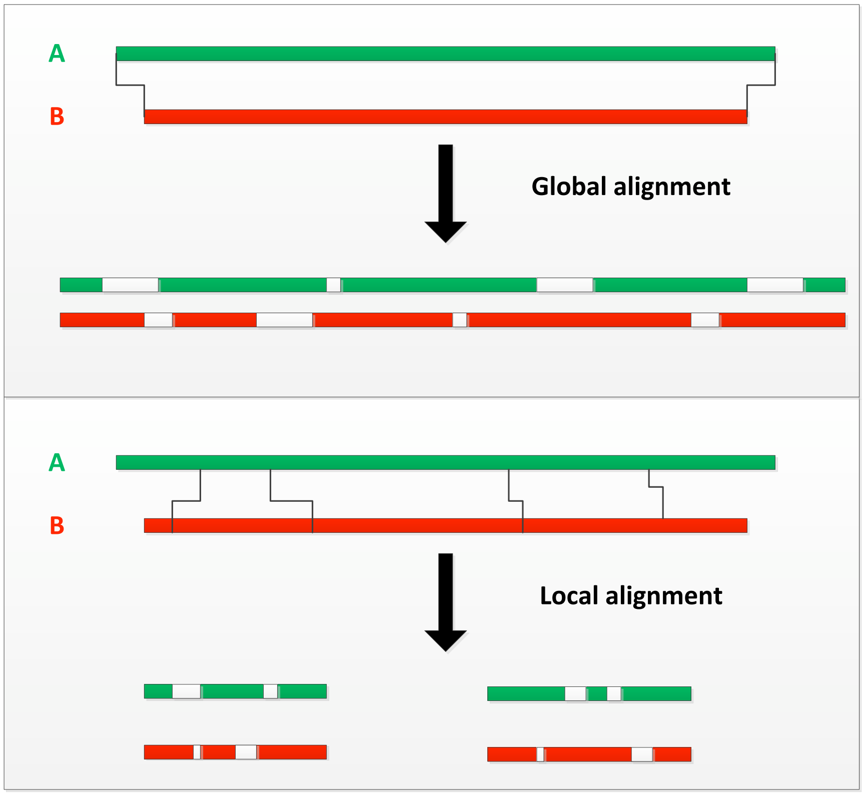


Figure 5. Types of sequence alignment. Global alignment tries to estimate how similar the two given sequences are, by stretching one sequence onto another, whereas local alignment searches for conserved regions.

1.3.2. Multiple sequence alignment and ClustalW.

While pairwise alignments can be used for tasks where we need similarity search, for building profiles and making phylogenetic analysis of related proteins we need multiple sequence alignments (MSA). MSA is alignment of three or more sequences (DNA, RNA or protein). Naïve way of constructing MSA is, first aligning two sequences with pairwise alignment, and taking the alignment as a single sequence, aligning it further to next sequences. While this approach is fairly simple to implement, it has been proven to be non-feasible for using to align large number of sequences. Using Big-O notation, which is commonly used for expressing computational complexity, this approach will take $O(L^N)$ time to construct where L is the length of resulting alignment in terms of number of residues, and N is the number of sequences aligned to each other. This means that the time needed for building MSA will increase exponentially as the number of sequences increases linearly. Under this conditions, finding global optimum was proven to be an NP-hard problem [18].

Since searching for guaranteed optimum solution in MSA building is prohibitively computationally expensive, new sub-optimal approaches were introduced which included heuristic steps. The most know of these methods are so called progressive alignments, which are not globally optimal. This method first calculates pairwise similarities between all the sequences, and using them, constructs a so-called *guide-tree* using clustering algorithms such as Neighbor-Joining [19] or UPGMA [20] . Then, it takes the most similar sequence pair as a starting point, and progressively extends the alignment by going from the next most similar to the most distant sequence, using the guide-tree. While this approach turned out to be fairly fast, it also has a systematic problem. If at any step there was made a mistake in aligning (simply because of the fact that at that point, misaligning gave higher score), this mistake is inevitably propagated to the rest of the alignment, worst case of which is the situation when the first pairs of sequences were aligned wrong, leading to a significant errors in resultant alignment. Nevertheless, this approach is the most widely used technique for building MSA. The best performing implementation of this approach is the Clustal family of programs [21], and among them ClustalW [22], [23] gained the reputation of the most popularly used one. ClustalW uses an additional level of heuristics, the most important of them being a process of assigning weights to partial alignments before starting to build guide-tree.

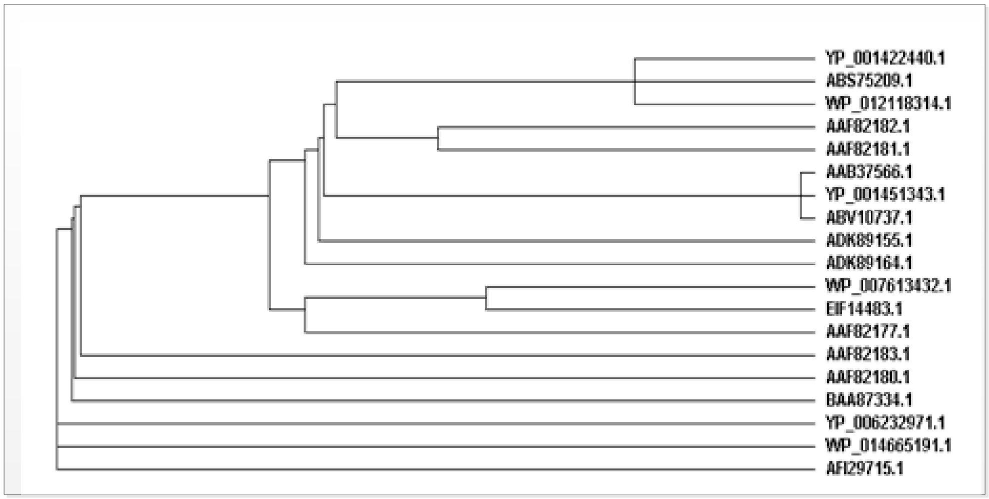
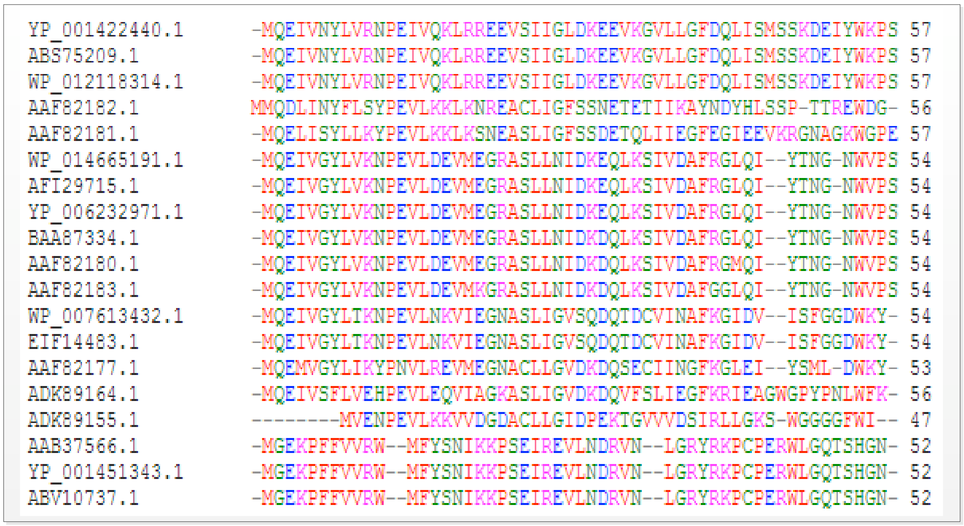


Figure 6. Guide-tree depicted as cladogram.
Tree produced by ClustalW using 19 *ComX* proteins from bacterial genomes.



```
YP_001422440.1      -MQEIVNYLVRNPEIVQKLRREEVSIIGLDKEEVKGVLLGFDQLISMSSKDEIYWKPS 57
ABS75209.1          -MQEIVNYLVRNPEIVQKLRREEVSIIGLDKEEVKGVLLGFDQLISMSSKDEIYWKPS 57
WP_012118314.1      -MQEIVNYLVRNPEIVQKLRREEVSIIGLDKEEVKGVLLGFDQLISMSSKDEIYWKPS 57
AAF82182.1          MMQDLINYFLSYPEVLKKLKNREACLIGFSSNETETIIKAYNDYHLSSP-TTREWDG- 56
AAF82181.1          -MQELISYLLKYPEVLKKLKSNEASLIGFSSDETQLIIEGFEGIEEVKRGNAGKWGPE 57
WP_014665191.1      -MQEIVGYLVKNPEVLDEVMEGRASLLNIDKEQLKSIVDAFRGLQI--YTNG-NWVPS 54
AFI29715.1          -MQEIVGYLVKNPEVLDEVMEGRASLLNIDKEQLKSIVDAFRGLQI--YTNG-NWVPS 54
YP_006232971.1      -MQEIVGYLVKNPEVLDEVMEGRASLLNIDKEQLKSIVDAFRGLQI--YTNG-NWVPS 54
BAA87334.1          -MQEIVGYLVKNPEVLDEVMEGRASLLNIDKDQLKSIVDAFRGLQI--YTNG-NWVPS 54
AAF82180.1          -MQEIVGYLVKNPEVLDEVMEGRASLLNIDKDQLKSIVDAFRGMQI--YTNG-NWVPS 54
AAF82183.1          -MQEIVGYLVKNPEVLDEVMKGRASLLNIDKDQLKSIVDAFGGLQI--YTNG-NWVPS 54
WP_007613432.1      -MQEIVGYLTKNPEVLNKVIEGNASLIGVSQDQTDCVINAFKGIDV--ISFGGDWKY- 54
EIF14483.1          -MQEIVGYLTKNPEVLNKVIEGNASLIGVSQDQTDCVINAFKGIDV--ISFGGDWKY- 54
AAF82177.1          -MQEMVGYLIKYPNVLREVMEGNACLLGVDKDQSECIINGFKGLEI--YSML-DWKY- 53
ADK89164.1          -MQEIVSFLVEHPEVLEQVIAGKASLIGVDKDQVFSLIEGFKRIEAGWGPYPNLWFK- 56
ADK89155.1          --------MVENPEVLKKVVDGDACLLGIDPEKTGVVVDSIRLLGKS-WGGGGFWI-- 47
AAB37566.1          -MGEKPFFVVRW--MFYSNIKKPSEIREVLNDRVN--LGRYRKPCPERWLGQTSHGN- 52
YP_001451343.1      -MGEKPFFVVRW--MFYSNIKKPSEIREVLNDRVN--LGRYRKPCPERWLGQTSHGN- 52
ABV10737.1          -MGEKPFFVVRW--MFYSNIKKPSEIREVLNDRVN--LGRYRKPCPERWLGQTSHGN- 52
```

Figure 7. An alignment of 19 proteins.
Produced by ClustalW using the tree depicted in Figure 6

1.3.3. Hidden Markov Models.

Hidden Markov Models (HMMs) are a mathematical framework used for building probabilistic learning algorithms for inferring meaningful conclusions from sequential data. While it was initially applied for speech recognition and signal processing, later the same approach was successfully ported to solve numerous biological problems, among which are gene prediction, protein structure prediction, sequence alignment and homology detection [24].

HMMs are similar to profiles, which were used extensively for homology detection before HMMs were introduced to bioinformatics. But instead of representing the profile as a two dimensional matrix, it uses a simple Bayesian network, where each position in the multiple sequence alignment, which is being modeled, is represented by three different states (insertion, match, deletion). For each of the states, the profile stores probability distributions for emitting for all possible residues. Moreover, once one state is processed, for passing to the next position in the sequence, there are transition probabilities to be issued for all three states. After setting the probability distributions, this HMM can be used to generate a sequence, just by following the path which gives the highest score. A sequence generated in this way is called *null model* and the score it gets going through the most probable path is used to normalize the score of a real sequence put on a test. When a sequence is subject to test for similarity, then it forces to go through states, which would produce the given sequence. Therefore, the states which would emit the given sequence are unknown until they are observed, hence they are *hidden*. This structure is represented by an oversimplified example in Figure 7.

The score of HMM search is calculated by the following formula:

$$S = \log \frac{P_m(sequence)}{P_\emptyset(sequence)} \qquad (1)$$

where $P_m(sequence)$ is the probability calculated by forcing the path based on evidence given by subject sequence, and $P_\emptyset(sequence)$ is the probability calculated from null model described above. Since the probabilities are multiplied along the path, they tend to result in extremely small numbers, thus it's more convenient to operate with *log* of the number. A score obtained in this way is also called *log-odds*.

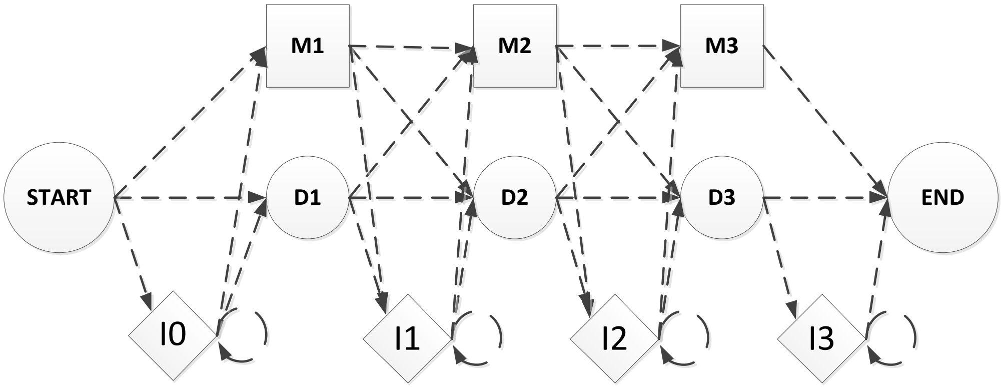


Figure 8. An HMM based profile. It models a multiple sequence alignment with length of 2. In case of protein sequence profiling, for each corresponding position in multiple alignment being modeled, there are 49 values stored: 9 state transition probabilities (the arrows), 20 match emission probabilities and 20 insert emission probabilities.

HMMs were applied for remote homology detection in several research groups and most of them were proven to be reliable [25][26][27]. Due to its convenience in use and high specificity, HMMs are today used widely in genome annotation projects, and databases like PFAM provide both multiple sequence alignments and HMMs for known protein domains [28].

1.3.4. BLAST.

While pairwise sequence alignment methods described in chapter 1.3.1 find optimal alignments, carrying a similarity search against large sequence databases with them can be extremely expensive in time and computational resources. To overcome this problem, numerous heuristic algorithms were developed. Among them, the most popular one is BLAST, developed by Altschul et.al at NCBI [29].

The BLAST (Basic Local Alignment Search Tool) is a sub-optimal aligner which reduced the running time of database searches in order of magnitudes comparing to optimal alignment methods. This improvement in speed was gained due to its underlying heuristic algorithm, the main idea of which is reducing the search space before starting doing full alignments. Algorithm works as follows:

1. Given a query sequence, every k-length word is produced sliding the window one by one until it reaches the end of sequence. (By default, *k* is 3 for proteins and 11 for DNA sequences.)

2. Every word is aligned with all possible k-length words. In case of proteins, for k=3, every word will be aligned with 20^3 3-length words. After that, only the words which gave a score above a certain threshold *t* are kept.

3. Having a set of words collected from the previous step, it searches for exact match throughout the database. Once the sequences having exact matches are found, the exact matching regions are extended in both directions until the time when alignment score starts to decline.

4. Having maximally extended aligned regions, BLAST looks for these regions within a distance *A* with each other. If this kind of region exists, they are merged together. These merged segments are called High Scoring Segment pairs (HSPs).

5. Calculates the similarity scores between HSPs and their statistical significances (e-values). The e-values are calculated using the following formula:

$$P(S \geq x) = 1 - \exp\left(-e^{-\lambda(x-\mu)}\right) \qquad (2)$$

$$\mu = \frac{log_2(K*m^{'}*n^{'})}{\lambda}$$

where . m' and n' effective length of query and database sequences, whereas the statistical parameters K and λ are found by fitting the scores, obtained by aligning query sequence and lots of shuffled versions of database sequence, to so-called *extreme value distribution*.

1.4. Types of genome annotation.

1.4.1. Structural genome annotation.

Structural genome annotation means identification of structural elements or segments in the genome. Finding protein coding genes is not complicated neither it is trivial. There are several methods for this task and most of them are based solely on the sequence. This type of genome annotation uses only the characteristics of sequence and is based on pattern recognition. The patterns that we are talking about maybe of many kinds, we might be looking for identities or we might use complicated regular expressions. A good example for complete identities between patters if ORF (Open Reading Frame) recognition which is based on recognizing the start codon (ATG) and the stop codons (TAA, TGA, TGG). When looking for a gene, we would like to find a series of codons, which starts with a start codon and ends with a stop codon. The only criteria we use here is that, the sequence should be sufficiently long. Thus, this is a simple exercise which we can implement with other patterns in a series that we recognize, simply based on codon characteristics. In practice, this method would be efficient enough provided that there are no sequencing errors. However a single sequencing error would destroy the process of recognition, so we need to have more complicated pattern recognition algorithms. In bacteria the program which is almost exclusively used is called Glimmer(Gene Locater and Interpreted Markov Modeler), developed by Steve Salzberg et.al [30] which was used extensively by the TIGR institute for annotating bacterial genomes on a large scale [31]. Glimmer is a system for finding genes in microbial genomes especially in bacteria, archaea and viruses. It uses Interpolated Markov Models to identify coding regions and to distinguish them from non-coding DNA. The

Interpolated Markov Model approach is described and published by Salzburg et.al in Nucleic Acids Research and it was later improved and published again in 1999 paper in Nucleic Acid Research. The glimmer principle is based on a combination of Markov Models and these models range from the first through the eighth order. And the order of data depends on the amount of data to train the model. Simple Markov Models, like the ones which are used here, are collection of transition probabilities which are based on observed statistics of sequences. In other words, these models are the generalized version of using codon tables, because instead of recognizing codons in all-or-none fashion, we are now recognizing mononucleotides, dinucleotides, trinucleotides etc. on a probabilistic basis. This approach works remarkably well for bacteria based on the fact that it interpolates between Markov Models of various degrees. When arriving at a coding sequence the model starts to look for three periodic non-homogeneous models which mean the generalized versions of the codon table. While this approach is completely adequate for bacterial genomes, it had to be complemented with other features for finding eukaryotic genes. The modified version of mechanism for finding eukaryotic genes includes HMMs and separate training procedures for splice sites, introns and exons. The details of eukaryotic gene finding will not be described since the eukaryotic genes are not in the scope if this study.

Altogether, glimmer became a complex recognition engine that can now adequately recognize various gene types. In bacterial section of NCBI's database, we find glimmer predictions for all annotated genomes. When we carry out structural genome annotation, in practice this almost invariably means the usage of one of the glimmer variants.

1.4.2. Functional genome annotation.

The term *functional annotation* refers to the process wherein we assign functions to hitherto unknown predicted gene. This gene prediction is based on structural gene prediction procedure described in previous section. In principle, the best method for functional prediction would be function determination which means experimental analysis of a gene based on gene knockouts etc. The advantage of this approach is that in principal we can recognize many new functions. However, given the variability of bacterial genomes, many gene functions cannot be determined using simple knockouts. The reason for this is that the behavior of knockout mutants is normally observed in the lab, under standardized growth conditions. And many of these specialized genes, and we know that bacteria are full of such genes, are simply not necessary for growth in those specialized conditions.

1.4.3. Homology based function prediction.

The other class of function determination is homology-based function prediction. This is the predominant way of function prediction today, which works well in bacteria. But it's hampered by the fact that each new bacterium has a large number of functions that may not be present in other instances of the same bacterium. When carrying out homology based function prediction, we take a predicted gene from a bacterial genome and compare it with the database of known bacterial genes and genomes. And if we find a significant similarity, then we can suppose that the unknown gene will have the same function. Such a similarity can point to same species, related species or a distant species. In practice, we see that the function of a gene can be conserved throughout quite different species and taxa.

Naturally this method has several problems. First, we may not know if a certain function is really represented in our database or it has a hitherto unknown role. This makes a problem in many cases where we have to determine whether or not a similarity is biologically significant. The fundamental problem is that even similar genes can carry different or similar functions. From this point of view, homologies can be divided into two groups: *orthologs* and *paralogs.* We consider two genes orthologous if they are in different species, carry out the same function and have evolved from a common ancestor. We call them paralogs if they are within the same organism, have the same ancestor, but carry out different functions. These functions can be completely different or different but related. (Figure 9) [32].

Finding orthology is the basis of homology base genome annotation and there are many more or less approved methods for distinguishing orthologs and paralogs. For a recent review see Kuzniar et.al [33]

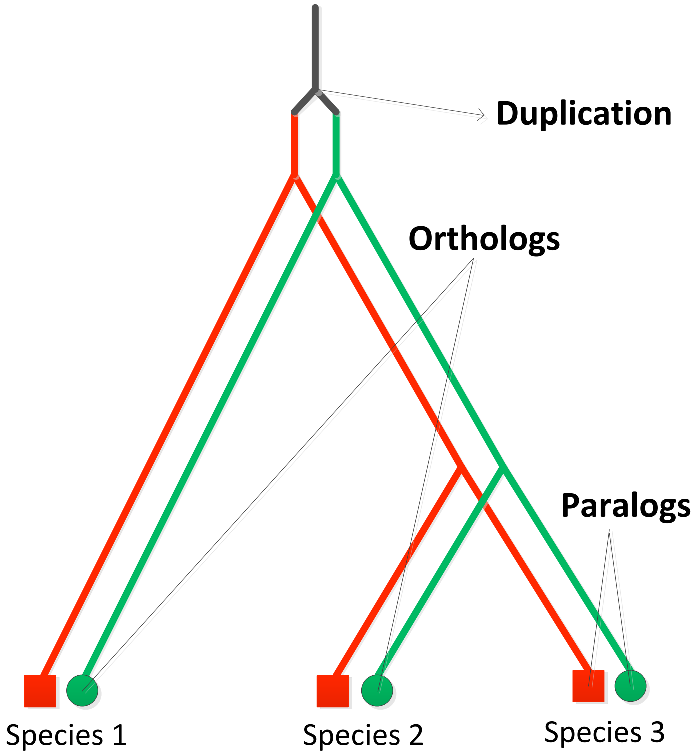


Figure 9. Homologous genes derived from a common ancestor. If they are in different species and carry out the same function, they are called orthologs. If they are in the same organism, but carry out different functions, they are called paralogs. The basis of evolution of paralogs is a gene duplication event.

1.4.4. Protein domain databases.

Protein domain databases are standard tools of genome annotation. Protein domains are complex three-dimensional structures in proteins, which can be well distinguished in three-dimensional (3D) structure of proteins and also can be well recognized based on the protein sequence alone. Protein domains are thus substructures of the entire protein and, as it is well known, they are evolutionarily autonomous units, which can hop from one gene into the other one, which have a well conserver exon-intron structure in eukaryotic proteins. But in bacteria, protein domains often constitute entire protein. Given the fact that protein domains can be very well recognized based on their 3D structures, there are several current collections of protein domains. The best known protein domain collection is the PFAM database, which was developed at Sanger institute in Hinxton [28]. The protein domain in PFAM is described by a Hidden Markov Model (HMM). This model is based on a multiple alignment which is also a part in the database. The PFAM database has two kinds of multiple alignments. One is the so-called SEED collection, which was used for developing HMMs. The complete alignment is also part of PFAM and it contains the elements of SEED as well as those which were not included in building the HMM but were recognized by HMM. PFAM is consolidated collection, which means that if they include something it will very probably be a true positive, since they prune the results, and they also cross-link such a domain description to external databases. The most important part of the external database is the PDB (database of protein 3D structures) which means that once you recognize a protein, you also have a picture of a representative example of that protein domain. In addition, which is equally important, PFAM also contains a well maintained description of that domain. This is like a mini review of that domain type, which describes its main functionalities, possible occurrences and various domain architectures in which this domain is part of. So PFAM, as of today, is a complex collection of information. Once we find a homology to a protein which exists in PFAM, we can link it to existing structure. From this context, it is interesting to note that PFAM was originally developed for protein domains. But bacterial proteins are so-called monodomain proteins so we have entire proteins structures as well. So now this is a complex database which contains information for domains mostly of known 3D structure. Also PFAM has various visualizing techniques, for instance the PFAM logo which is shown in Figure 10 represents the residue conservation in protein domain.

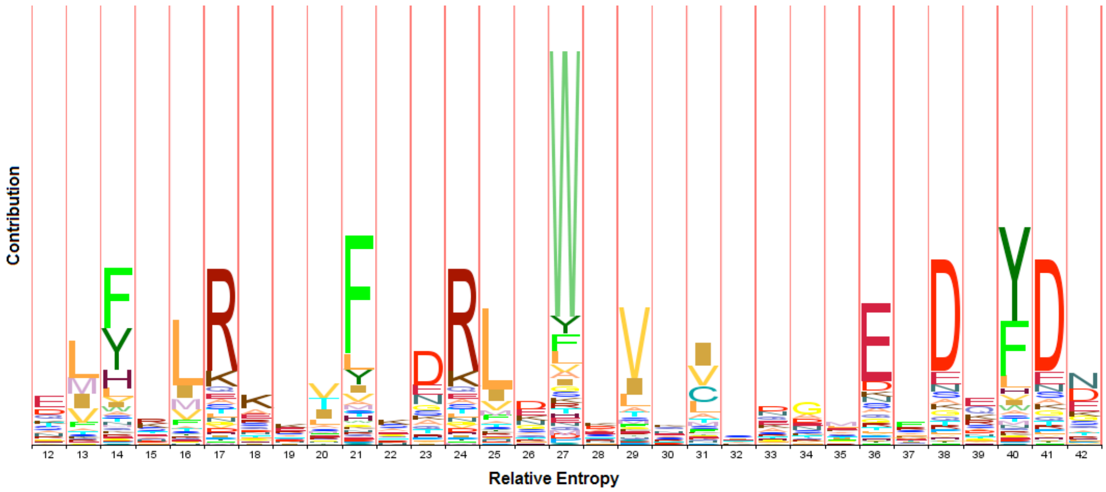


Figure 10. The logo of PFAM a family PF00765. It is an auto-inducer synthase protein. The size of the letter and each position is proportional to its conservation level which is calculated as relative entropy.

It is important to note that, in the beginning, protein domains were first described in terms of regular expressions. This was so-called PROSITE database [34]. The PROSITE database was the first to determine residue syntax for describing protein sequences and used regular expressions for describing the conserved protein segments which were the protein domains. This database was later complemented with profile-type descriptions of proteins that also included relative frequencies in various sequence positions [35][36], which are in fact very similar to HMMs, being in fact equivalent. The reason for including profile-type description was that it was obvious from the beginning that conservation cannot be described in terms of yes/no occurrences which are basis of regular expressions. Another early approach which was based on this recognition is so-called SBASE approach (or domain library approach) [37]. In this approach we do not develop a consensus such as an HMM or a sequence profile, but we represent a domain type as a collection of sequences. In order to maintain such a collection, we don't need multiple alignments which actually require a large human overhead. SBASE was the first publicly available domain sequence collection and it was later complemented by literature reviews and today is not updated anymore. Doubtless advantage of this approach is that it can recognize also atypical domain instances which are missed by HMMs and profiles. The reason for this is that HMMs can be biased towards one type if the sequences of that type are predominant in sequence collection used for making multiple sequence alignment for building that HMM. SBASE does not have this drawback but it is based on sequence similarity searching which is less sensitive than HMM searches. So this approach is still viable and it can complement HMM searches.

1.5. Functional databases.

For describing and organizing functional descriptions we have two derivative database types. First type is a collection of sequences grouped according to protein functions. Archetype of this database is COG [38]. A different type of database is actually meta-database which stores only a standardized description of functions organized into a concept hierarchy and a set of rules which is often called *ontology*. The best known ontology in bioinformatics is GO (Gene Ontology) which is maintained by the Gene Ontology Consortium [39].

1.5.1 The COG database: Clusters of Orthologous Groups.

The COG database [38], is the first and perhaps the best known protein sequence database used for annotating functions in bacteria. A COG group is a cluster of sequences that are mutually similar to each other above a certain threshold of BLAST similarity. Such a group contains supposed orthologs and evolutionary relatedness was validated by human operators. The database was published first at the end of 90s on experimental basis so as to facilitate the annotation of bacterial genes. Based on the early experiences, they defined 17 main functional classes in which most of the biological functions were known. But based on sequence similarity, there were also COGs with unknown function. The functions were grouped into higher classes like energy metabolism, information processing etc., which resulted in a very well structured and not too complex concept hierarchy. We note that this concept hierarchy is different from that of the GO database today but it is still used. The database was very successful and it has grown to close to 5000 groups of bacterial proteins close to ¼ of which were genes with unknown function [40]. Unfortunately NCBI has decided to discontinue the development of COG, so database is no longer updated. Nevertheless it is still used almost inevitably when annotating bacterial genomes.

There are several automated databases such as eggNOG [41] ,which were devised to take the role of COG database. But as these databases do not include human curation, they are far less popular in automated genome annotations, which are still mostly based on COG database.

1.5.2. Gene ontology

The Gene Ontology (GO) database contains a standardized description of gene functions [39]. Originally this database was developed for description of *Drosophila* genes but the authors of this database have recognized that there are no other standardized systems of concepts for other biological domains, so they extended their work to bacterial and eukaryotic genes as well. Today GO is a general system which is used throughout all biological kingdoms. GO is an ontology, so it contains a system of concepts organized into a directed acyclic graphs which are similar to concept hierarchies used in simpler systems, but they also allow one concept to belong to several parent concepts. The advantage of GO is that logical integrity of concepts can be automatically validated. And once this is done, an accepted function description in GO can be used in functional annotation and will allow to eliminate misunderstandings and chaos which was always the case when unstandardized names were used. An example is shown in Figure 11.

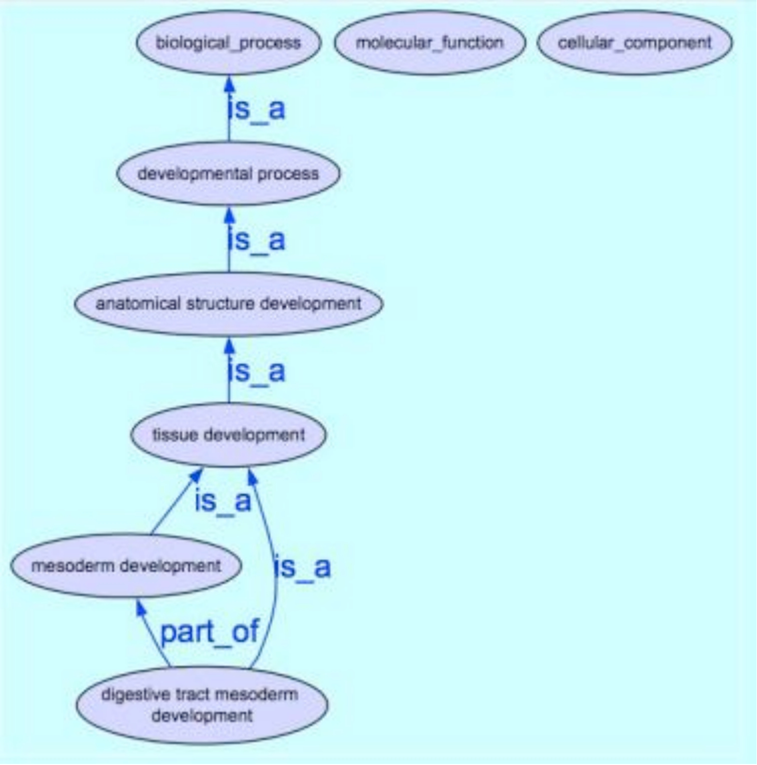


Figure 11. Gene Ontology. The edges of acyclic directed graph, which is used as a data format in GO, represent type of relationships between the ontologies (the nodes in the graph). A relationship can be: *is_a*, *part_of*, *regulate* and *has_part*. For example, in the figure above, "Mesoderm development" is a "tissue development", which means that former is the subtype of latter. Similarly "Digestive tract mesoderm development" is a part of "mesoderm development" process. When a linked lineage of ontologies (a branch of the tree) is traceable from leaf node to root following only *is_a* relationships, this ontology tree is called "*is_a* complete".

1.6. Bacterial communication.

Bacteria communicate with each other using secondary metabolites, to coordinate their actions, behaving like multicellular organisms. This behavior can be well understood in evolutionary context, since it enables bacteria to better exploit the resources and adapt to environment [42].

1.6.1. Secondary metabolites

Secondary metabolites are organic molecules produced by organisms, which do not play primary role in normal growth, development and reproduction. Absence of a secondary metabolite or malfunctioning of the secondary metabolite producing system does not cause an immediate death or effect on organism. This was the subject of a debate on real functions and importance of secondary metabolites [43]–[45]. A secondary metabolite is usually used by one or a small group of species. Some secondary metabolites function as a communication signaling tool among producer organisms (plants, animal and microorganisms) which share the same environment.

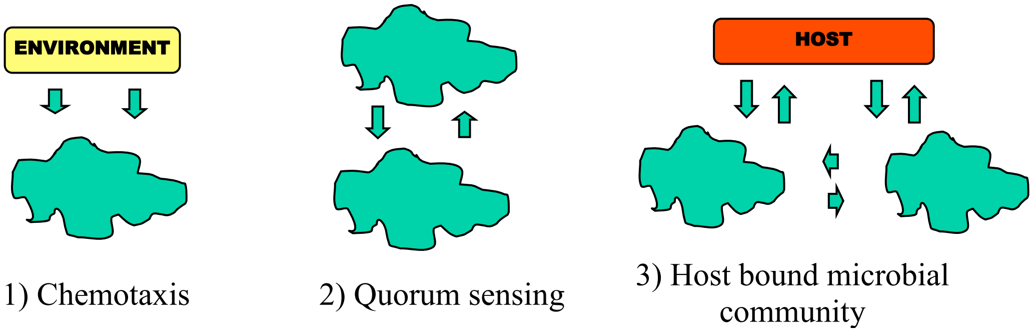
Microorganisms release a large number of secondary metabolites and they also take up secondary metabolites released by their neighbors. In this manner, they both react to as well as modify their environment.

1.6.2. Quorum sensing

Quorum sensing (QS) is a type of cell-to-cell communication system where bacteria react to fluctuations in cell population density by regulating the expression of specific genes [46]. This process is carried out by using secondary metabolites called autoinducers. When the level of autoinducer concentration reaches a certain threshold level, the bacteria population reacts by a synchronous expression of response gene [47]. In this manner, the concentration of autoinducers in the environment is used as a measure of population density. This behavior makes it possible for bacteria to both survive and change hostile environment.

An example to this can be a study done by Chandler et.al. using *Burkholderia thailandensis* and *Chromobacterium violaceum* as model bacteria. Both of the bacterial species use QS to trigger production of antibiotics, which will inhibit the population growth of other species. The study showed that the signal receptor encoded by *Chromobacterium violaceum* can sense the signals produced by *Burkholderia thailandensis*, thus making it possible for the former to eavesdrop on later. This gives *Chromobacterium violaceum* competitive advantage in certain cases [48].

From the theoretical point of view, QS is a particular subcase of a cell responding to extraneous molecules adverted by diffusion from the environment (Figure 12). In this general sense, the cell can simply respond to a molecule in the environment, such as is the case in chemotaxis [49]. In chemotaxis a cell will follow a concentration gradient of a molecule within the environment. QS is a particular case wherein the molecule to which the cell responds is produced by the cell itself. This subcase is also called *autocrine signaling*. Finally, microbial communities living inside a host have a complex network of interactions. First they respond to molecules produced by the host, and also produce molecules that produce a response in the host. This is a subcase of *paracrine signaling* wherein different cells interact with signals they themselves do not necessarily react to. Within a community of host bound microbiota cells of the same population may also communicate via QS (autocrine signaling), and they also may communicate with each other via antimicrobial factors, etc., which is a case of paracrine signaling.



1) Chemotaxis

2) Quorum sensing

3) Host bound microbial community

Figure 12. Cases of bacterial signaling. 1. Microbes, such as bacteria have a large number of sensing systems that allow them to respond to molecules of the environment. Example: Chemotaxis. 2. Microbes also respond to their own signals which allow them to sense their neighbors, to share public goods and to form complex communities. Example: Quorum sensing. 3. Interactions with host organism can build complex microfloras called microbiota or microbiomes. Example: Gut flora

Interestingly, the signaling mechanism between cells is based on very similar principles both on prokaryotes and eukaryotes. Typical examples are the so-called two-component systems of bacteria [50] which are very abundant, some species can have more than hundred such systems, all presumably responding to different materials. In a two-component system, the extracellular signal is bound to the extracellular part of a transmembrane receptor. A conformational change occurs in the receptor, and the internal part, a kinase enzyme becomes active. The kinase will phosphorylate the second component of the system, a response regulator, which will bind to DNA in the nucleus triggering a change in gene expression. This elaborate system allows a signal being transduced through the cell wall, without any molecule passing through the cell wall. Prokaryotes that have strong peptidoglycan cell walls such as Gram-positive bacteria use this kind of two-component system also for QS. In *Bacillus subtilis*, the response regulator *comA* activates the production of a peptide signal *comX* and a transport protein *comQ* that binds to the internal membrane. *ComQ* then processes the *comX* peptide molecule, in two enzymatic steps, modification by isoprenylation and cleavage [51] . The resulting product, the mature, isoprenylated *comX* peptide is released from the cell. This case is an example of active transport which ensures that only specific molecules can pass through, and there is no diffusional contact between the cell interior and the external environment (Figure 13 B). A simplified version of this system is used by Gram-negative bacteria that do not have a peptidoglycan cell wall so their cell wall is more penetrable by small molecules. The main difference is that in simple Gram negative systems a) the signal molecule can pass through the cell wall by diffusion, so the external and internal signal concentrations are in equilibrium, and b) The signal directly binds to the response regulator protein which will then bind to DNA. In this system, the signal is a small molecule, synthesized by a signal synthase that is activated by the signal itself in a process called autoinduction. (Figure 13 A)

The fundamental steps involved in the response to fluctuations in cell number are similar in all QS systems. In a canonical AHL system, the autoinducer molecules are passively released or actively secreted outside of the cells. As the number of cells in a population increases, the extracellular signal concentration likewise increases and when it accumulates above the minimal threshold level required for detection, cognate receptors bind the autoinducers and trigger signal transduction cascades that result in population-wide changes in the gene expression (Figure 14).

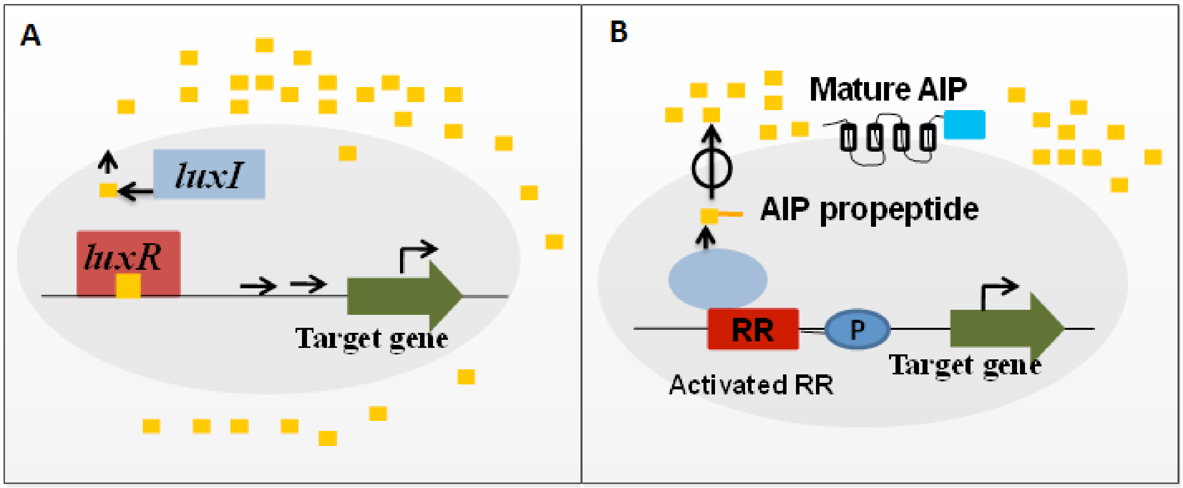


Figure 13. Types of signal transduction. A) One component system: small signal molecules penetrate the cell membrane freely, leading to balance in concentration of signal molecules inside and outside. B) Two component system: Signal is passed using a trans-membrane protein, to which the signaling molecule binds only from outside of cell membrane.

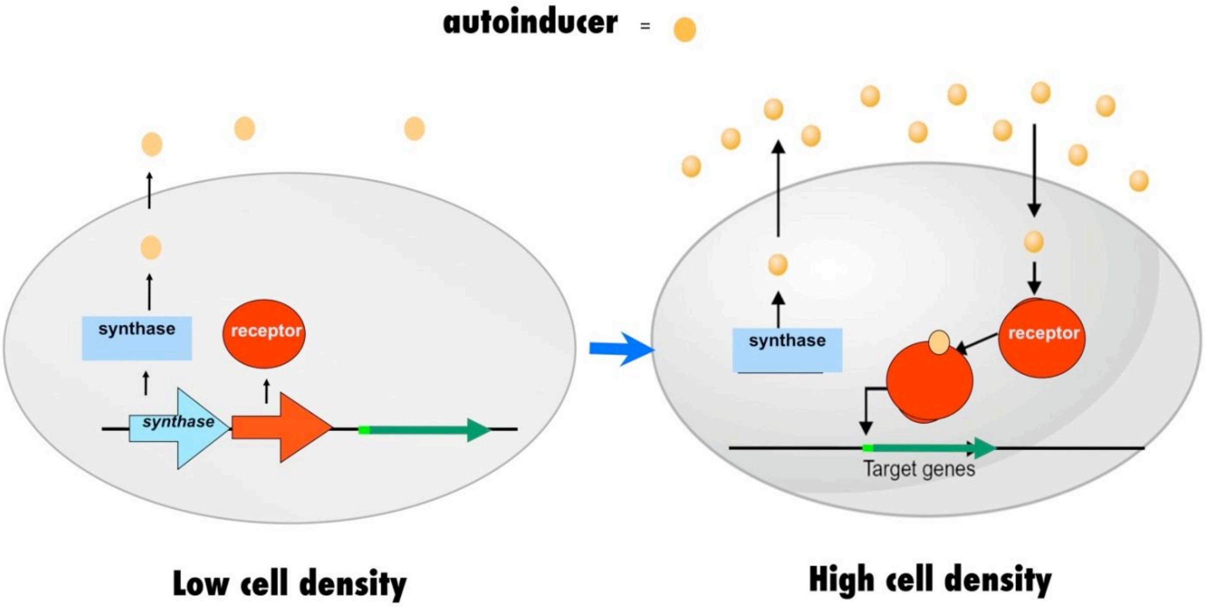


Figure 14. A canonical example of QS machinery. Autoinducers are produced for making it possible to measure the cell density. Once the cell-density reaches a specific threshold level (high autoinducer concentration), the autoinducer molecules get bound by receptors which will then lead to alterations in gene expression level.

Well known and most studied autoinducers are the *N*-acyl homoserine lactones (AHL), while other autoinducers used for bacterial communication are also known (such as oligopeptides).

The genes involved in QS regulation tend to be located in well-organized clusters on the chromosome, classified by Goryachev into three types [52], [53]. These clusters usually include the genes for encoding enzymes which are responsible from synthesis of response molecules, as well as the genes which provide resistance to some toxic secondary metabolites (such as antibiotics) [54]. This nature of genes which encode proteins/enzymes which constitute QS machinery makes it suitable to use subsystem based approach to annotate them.

However, there is a large number of *luxR* genes that are not paired with *luxI* genes hence they are often called orphans [55] or solos [56]. The overall architecture of the encoded polypeptides is highly similar to QS-linked *luxR* proteins, so it is very likely that solo *luxR* proteins respond to signals that in principle can be intracellular or extracellular. Among the intracellular signals one can think about the signal of a distantly related *luxI* gene as is the case with *P. Aeruginosa* [57], but it can be any intracellular metabolite. Among the extracellular signals one can think of environmental cues, such as those involved in chemotaxis. A very interesting hypothesis by Venturi and associates proposes that plants recruit their symbionts as well as the members of their rhizosphere via chemical signals perceived by solo receptors [56]. Detailed modeling experiments showed that the active pocket of *luxR* solos may have evolved to accommodate such foreign, non-AHL molecules. From the regulatory perspective, the case of solo *luxR* proteins responding to foreign signals is not unique. One of the goals of my thesis is to make a census of solo *luxR* genes/proteins in the current genomic databases.

This study is mainly concentrated on annotating QS genes of Gram negative bacteria, and studying their characteristic topological arrangements, using various bioinformatics tools.

1.7. Horizontal gene transfers

1.7.1 Theory

Horizontal gene transfer (HGT), also called lateral gene transfer (LGT) , is said to occur between organisms when the donor of the genetic material is not the parent of the acceptor [58]. For HGT to occur between two bacteria, the cells have to be in physical contact. Once the transferred DNA is incorporated into the genome, it is then “vertically” inherited from parent to offspring. The molecular mechanisms of HGT had been studied since the 1940’s but it was only after the appearance of the first complete genomes in the 90’s when it was recognized that HGT is a major factor of microbial evolution. Today HGT is considered as key to many important processes in the microbial world, such as, for instance, the spreading of bacterial antibiotic resistance [59]–[61]. In general, HGT is very useful for environmental adaptation, better than point mutations. Dense microbial communities, such as the human gut microbiota are generally considered as a hot spot of microbial gene transfer [62]. This is all the more significant since it was recently discovered that the rate of HGT is apparently eight to nine orders of magnitude faster than previously thought [63]. As a result, rapid microbial evolution is now believed to be a major factor that can shape the community structure of microbial consortia [64]–[66].

In bacteria, HGT has three main avenues.

- Transformation – some bacteria (“transforming” bacteria) can take up short pieces of naked DNA, this is a commonly occurring mechanism.

- Transduction – phages can transport DNA together with their own genomes. The length of DNA is limited by the carrying capacity of the phage head. Donor and recipient have to be closely related since they must share cell surface receptors.

- Conjugation-plasmids/transposons, cell to cell contact, distant relations, long DNA.

The requirements of gene transfer include many factors, such as proximity of the acceptor to donor DNA, stability of DNA in environment, vector transmission, uptake/insertion mechanisms as well as proper maintenance, stabilization and selection in the new host. Factors limiting or preventing HGT include instability in the new host, restriction systems that eliminate foreign DNA, GC/Codon usage incompatibility as well as the lack of appropriate interacting genes. As a result of these factors the entire scenario of HGT can be pictured as follows: DNA arriving into a host cell first carries all characteristics of the donor genome. If these characteristics are not compatible with the acceptor, the incoming DNA will not survive. If it is successfully incorporated into the acceptor chromosome, it will carry the original characteristics of the donor genome, however these characteristics (often called “signatures”) will slowly mutate away and the new DNA will become more and more similar to the host genome. This simple view implies that recent HGT events can be more easily spotted, also HGT between genomes widely differing in their local characteristics can be better detected and will be detectable for longer times.

Known instances of HGT include:

- Antibiotic resistance genes on plasmids
- Insertion sequences
- Pathogenicity islands
- Toxin resistance genes on plasmids
- Agrobacterium Ti plasmid
- Viruses and viroids
- Organelle to nucleus transfers

HGT is studied in two broad contexts: i) In genomics, the question usually is to pinpoint regions of a chromosome that are likely to be newcomers as compared to the rest of the genome. ii) In functional studies, on the other hand, we want to decide if a given gene (gene family) has a tendency to undergo HGT, where it arrived from, etc. In both cases, the analysis is predictive and we can only suggest that HGT is likely reason. Early genomics studies indicated that the percentage of foreign DNA can be quite substantial in bacterial genomes:

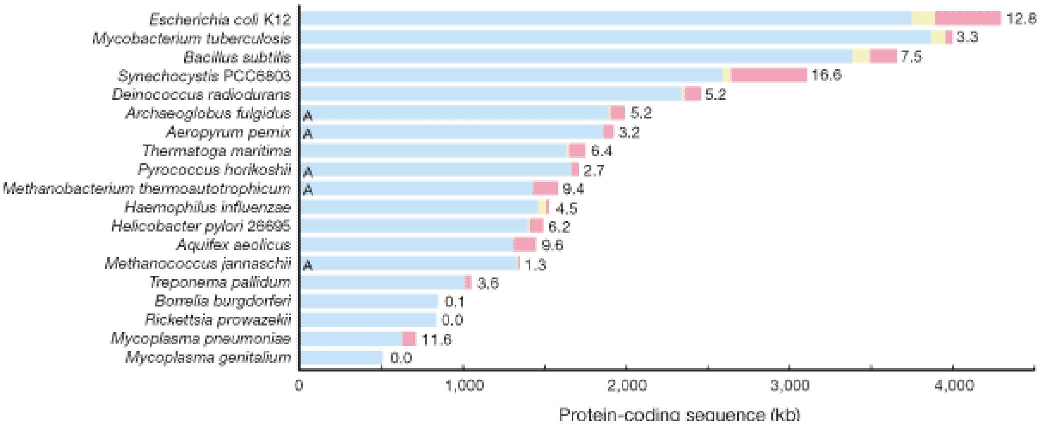


Figure 15. Percentage of foreign DNA in bacterial genomes, from [67]. Length of bars represent the amount of coding DNA, native is blue, foreign due to mobile elements is yellow, other is red. Numbers indicate predicted percentage of foreign DNA.

1.7.2 Principles of computational analysis

The methods of HGT prediction fall into two categories: a) phylogenetic analysis. b) sequence characteristics, also called parametric analysis, of which GC content is the classical example.

a) Phylogenetic methods – that will not be described here in detail – require the comparison of phylogenetic trees constructed from various genes. In bacteria, the gold standard is 16S RNA gene which is most often used to classify bacteria. HGT can be suspected if the phylogenetic tree of a given gene is obviously different from the 16 RNA tree. Naturally, HGT is more probable if a series of adjacent genes show the same phylogenetic anomaly.

The simplest, most practical question is to decide if a gene is rare or unique within its taxonomic neighborhood. A gene that is present in a single bacterial strain within a species or a genus, is likely to have arrived by HGT, if similar genes are frequent or obligatory in other taxa. This probability is corroborated if the two suspected organisms live within the same environmental niche. The phylogenetic approaches are summarized by Azad and Lawrence, 2012 [68] .

b) Parametric methods are based on determining a local characteristic of a genome segment. Plotting this value as a function of sequence position within the genome allows one to pinpoint conspicuous regions that may correspond to alien DNA. Such outlier regions are usually identified by numerically comparing calculated metric of a local segment to the globally calculated value of the same metric of the same genome, or to the immediate environment of the segment analyzed.

G+C content. The earliest example of parametric methods is G+C content (from Lawrence and Ochman (52–54), from book chapter). As G+C content (also written as GC content) has a profound effect on DNA stability, it widely varies between bacteria. Also, high GC genomes have a tendency of having genes of both DNA strands as opposed to low GC genomes, where one of the strands contains most of the genes. So if a genome has an overall GC content of 52%, a segment with only 40% can be easily spotted by plotting the GC content along the chromosome. Obviously, if there is little or no difference between the GC content of the donor and acceptor genomes, the segments cannot be easily spotted. From the numerical point of view, GC plots are noisy curves with a lot of local fluctuations which can obfuscate HGT events, especially those of shorter DNA segments. GC plots are an example of window-sliding algorithms: GC content is calculated for a given sequence window which is then slid along the sequence. This is perhaps the oldest and simplest algorithm types in bioinformatics and many signal processing tricks known in the engineering literature can be applied to them [69] . As a consequence, the methods of GC analysis are quite varied even though many of methods differ only in slight technical details (overview of the commonly used methods is in book chapter by Azzad and Lawrence [68]).

The picture of GC analysis can be easily widened, and in several directions. The main directions are as follows.

Varying the alphabet size and vector content. GC analysis describes sequences in a two-letter alphabet (G or C correspond to one, A or T to the other character) so the composition of a window is described with one single number. We can use larger alphabets, for instance there are 16 dinucleotides, 64 trinucleotides etc. so that the window composition will be described with the vectors of 16 or 64 dimensions, calculated as the frequency of overlapping di- or trinucleotides, respectively. We can reduce the alphabet size by considering reverse complements of di or trinucleotides as equivalent. Today, tetranucleotide descriptions are generally used, larger nucleotides are rarely used, simply because they are less and less frequent in genomes as we increase the size, so the resulting vectors are sparser and sparser, especially for shorter sequence windows. An interesting example of nucleotide signatures are the relative abundance measures *f'*, defined by Karlin et.al. [70][71], defined for a dinucleotide word as follows:

$$f'(AT) = \frac{f(AT)}{f(A) * f(T)} \quad (3)$$

where $f(A)$ and $f(T)$ are the frequencies of A and T, respectively, $f(AT)$ is the frequency of the word AT. By extension:

$$f'(ATC) = \frac{f(ATC)}{f(A) * f(T) * f(C)} \quad (4)$$

$$f'(ATCG) = \frac{f(ATCG)}{f(A) * f(T) * f(C) * f(G)} \quad (5)$$

etc. *f'* vectors have the same number of dimensions as f vectors, but are thought to be more sensitive.

An interesting type of vector descriptions are those that take into consideration the structural equivalence of nucleotide words. Briefly, a nucleotide word on the positive strand strictly corresponds to a reverse complementary nucleotide word on the negative strand. Strict correspondence means that if one of them occurs, so will the other one. The databases contain only one strand of the genome, but in principle there is no reason to leave the other strand out of the calculation. The rule of reverse complementarity is schematically shown in Figure 16

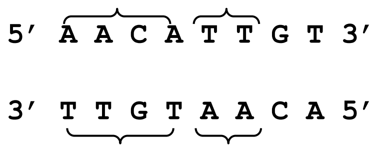


Figure 16. Structural equivalence of reverse complementary words. Note that AAC on the upper strand is equivalent with GTT on the lower strand. But the word AT is the same on both strands.

The important property of structural equivalences (SE) is that they decrease the dimensionality of the vector descriptions. Instead of 64 trinucleotides we will have 32 structurally equivalent trinucleotide pairs. We can make SE descriptions for mono, di, tetranucleotide etc. descriptions. It is interesting to note that the SE version of mononucleotide descriptions is the GC content itself. Another interesting point is that the dimension reduction is different for even numbered nucleotide words. For instance, the word AT is the same on both strands which means that the 16 dinucleotides correspond to 10 SE dinucleotides. This principle is widely used in DNA structure prediction (for instance see Brukner et.al. [72]), but there is no systematic study on the genomic data available today.

Using more than one parameter. We can compute more parameters for characterizing a sequence segment, for instance we can use GC content and tetranucleotide composition. One possibility, used early on, is to plot the codon usage of the ORFs within a given region. Naturally, codon usage can only be calculated for coding regions, and it is known to strongly vary with the expression.

The parameters that can be used are not restricted to composition-like features, we can use physicochemical parameters (e.g. melting point), computed parameters (DNA curvature) etc. The plot.it server [73] offers 45 parameters for DNA that can be combined in various ways. The most typical method is to use more parameters is to make multiple plots, and visually identify peak/valley regions. Another, widely used method is to make a scatter plot using two parameters for each window and then study the clustering patterns either visually, using a 2D plot, or with any method of cluster analysis. Further, we can make histogram-like representation, e.g. 3D plots etc.

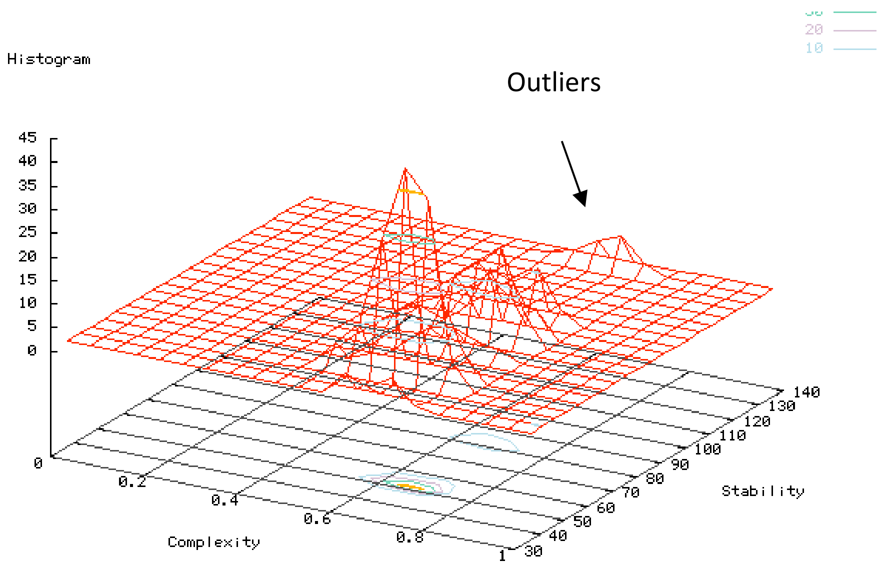


Figure 17. Finding outliers in multidimensional plots.

Vector comparison measures. The key of all calculations is a numerical measure of comparison. Two vectors, for instance, one describing the sliding window at a given position, and the other one describing the genome, can be calculated with any of the vector distance measures, or such standard similarity measures, such as the dot product of the two vectors. For comparing two tetranucleotide frequency vectors the dot-product is

$$S_{1-2} = \sum_{1}^{256} f_i^1 * f_i^2 \qquad (6)$$

which is by definition symmetrical and within the range of [0,1], the latter condition if fulfilled if the f vectors are normalized to the sum of vector.

Today, the comparison is often carried out with the Kullback-Leibler (KL) divergence measure. The original KL measure for two (for instance tetranucleotide) vectors f_i^1 and f_i^2 is

$$KL_{1-2} = \sum_{1}^{256} f_i^1 * \log \frac{f_i^1}{f_i^2} \quad (7)$$

whereas the symmetrized measure is

$$KL_{sym} = \frac{KL_{1-2} + KL_{2-1}}{2} \quad (8)$$

also called as Jensen–Shannon divergence.

Outlier search. The parametric methods mentioned so far were based on plotting a measure calculated from local characteristics as a function of sequence position. If we compare this value with global characteristics of the genome, we get difference plots where only outlier regions show up as peaks or valleys. Typically, we calculate a vector for the window and a vector for the entire genome, and compare these in terms of the KL_{sym}. These plots allow one to pinpoint outlier regions, Figure 18 (left). Naturally, thresholds are needed to select the significant peaks. But if a long segment of the genome has an above average KL value, this may be indicative of HGT than simple isolated peaks. This is a fairly simple principle but it has a tacit assumption that is rarely mentioned: By comparing local and global values we assume that a local property has to be similar to a global property. G+C content and nucleotide signatures may be such properties, but there is little theoretical backing to this, so outlier regions do not necessarily indicate HGT events.

Current methods use a combination of the above principles. A good example is the web server of Dufraigne et.al [74] that uses tetranucleotide signatures for describing genomes and KL divergence as a distance metric. The tetranucleotide signatures are compositional vectors that are cleaned from the influence of alien DNA. To do this, the genome is divided into overlapping segments and a – somewhat arbitrary, k-means based – clustering algorithm is used to identify the outliers which will then be omitted when the tetranucleotide vector (signature) is calculated. This gives rise to a set of genome vectors and the segments of the query DNA are compared to this set.

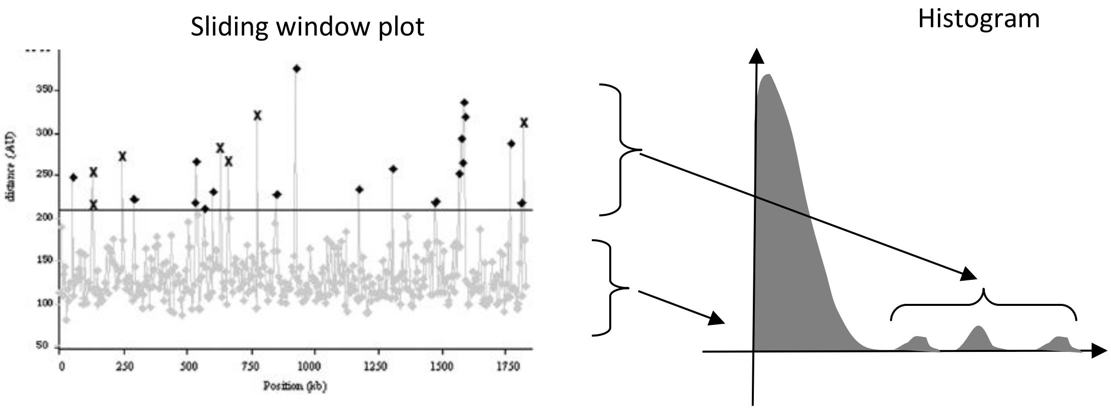


Figure 18. Identifying outlier regions in DNA. *Left*: A Kullback-Leibler plot of a genome. *Right*: A histogram of the KL values obtained for the individual windows.

Another successful example of server for detecting HGT is GOHTAM server, developed by Ménigaud et.al [75], which is an improved version of the server done by Dufraigne et.al [74], described above. Differing from the methods described above, the GOHTAM server uses KL_{sym} (Jensen–Shannon) divergence instead of plain KL divergence. The main feature of this server is that it can combine parametric and phylogenetic methods, giving as an option for a user to choose, whether to do them both or only one of them. It stores the signatures of not only the complete genomes, but all bacterial GenBank entries, longer than 1 kb. And when user submits a genomic segment (or GenBank entry) for analysis, it calculates and brings up 10 nearest neighbors in terms of signature distance. The set of items that GOHTAM gives is as follows:

- *Potential source genomes*. It gives 10 nearest neighbors found from the database, in terms of tetranucleotide signature distances, as described above.

- *Phylogenetic trees* built using the signature distances in neighbor joining algorithm, as described in [76].

- *Oligonucleotide content*, which is a visual representation of signature as a matrix of dimensions 16x16, obtained from vector of oligonucleotide frequencies of size 256 (since it works with tetranucleotide signatures), as described in [77]

- *Genome alignment* of input sequence to the genomes of possible origin, using maximum unique matches (MUM) approach as described in [78].

It is worth mentioning that today there are a number of further web servers that can evaluate genomes for HGT and databases that assign pre-computed HGT probabilities to the genes of microbial genomes. Also, there are number of ready-made programs that can be used in microbial genomics pipelines.

Some of the servers and packages for predicting horizontal transfers or other useful information:

Name	Location	Features
Trex	http://www.trex.uqam.ca	Phylogenetic inference and visualization [79]
Horizontal Gene Transfer database	http://genomes.urv.es/HGT-DB/	Database of known horizontally transferred genes[80]
Alien hunter	http://www.sanger.ac.uk/resources/software/alien_hunter/	Detection of putative HGTs using Interpolated Variable Order Motifs [81]

Table 1. Online tools for horizontal gene transfer detection.

Some tools for comparative genomics:

Name	Location	Features
VISTA	http://genome.lbl.gov/vista	Suite of programs and databases for comparative analysis of genomes [82]
ACT: Artemis Comparison Tool	http://www.sanger.ac.uk/resources/software/act/	Displays pairwise comparisons between two or more DNA sequences [83]
SyntTax	http://archaea.u-psud.fr/SyntTax/	Tool for linking genomic elements according to their taxonomic relations. [84]
LAST	http://last.cbrc.jp/	Finds and displays similar regions between genomic sequences[85]
CoGe	http://genomevolution.org/CoGe/	Suite of tools for comparative genome analysis[86]

Table 2. Online tools for comparative genome analysis.

1.7.3 Gene and operon comparisons

The philosophy of HGT testing is seemingly very different when well defined segments of genomes are analyzed. As a hypothetical example, suppose we want to find out if an operon $O_{pseudomonas}$ of *Pseudomonas aeruginosa* comes from *Burkholderia cepacia* where it is called $O_{burkholderia}$. We calculate vectors for the operons as well as for the genomes, $G_{pseudomonas}$ and $G_{burkholderia}$. We also need an "average genome" that can be of a strain which is not related to *P. aeruginosa* or *B. cepacia* but is of similar GC content. We can test the hypothesis of *O* coming from *Burkholderia* by comparing the vectors of all operons and genomes with each other, in terms of a distance measure such as the symmetrical Kullback-Leibler divergence KL_{sym}. We can then support the hypothesis in terms of conditions written in the form of inequalities:

$KL_{sym}(O_{pseudomonas}\ ,\ O_{burkholderia}) < T_1$ (very low) (i)

$KL_{sym}\ (O_{pseudomonas}\ ,\ G_{burkholderia}) < T_2$ very low or low (ii)

$KL_{sym}(O_{pseudomonas}\ ,\ G_{pseudomonas}) \sim KL(O_{pseudomonas}\ ,\ G_{average}) > T_3$ (iii)

high or significantly higher.

$KL_{sym}(O_{burkholderia}\ ,\ G_{burkholderia}) < T_4$ *(very low or low)* (iv)

where T_i are threshold values. These inequalities express the common sense expectation that a segment should be similar to its donor sequence and to its donor genome, but less similar to its own genome. And the respective operon in the donor genome is similar to the donor genome itself. Note that the inequalities symmetrically change if the operon was transported from *Pseudomonas* to *Burkholderia*. Such clear pictures are rarely obtained, because, same as stated for GC content above, there are problems if there is little or no difference between the vector descriptions of the donor and acceptor genomes. In order to make the evaluation more robust, the above inequalities are often simultaneously tested for a few descriptions, such as tri- or tetranucleotide signatures as well as codon usage vectors etc. Usually, this analysis is manual as given in Figure 19.

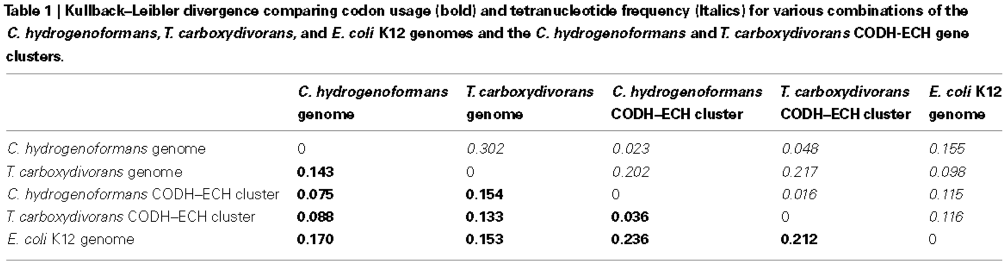


Table 1 | Kullback–Leibler divergence comparing codon usage (bold) and tetranucleotide frequency (Italics) for various combinations of the *C. hydrogenoformans*, *T. carboxydivorans*, and *E. coli* K12 genomes and the *C. hydrogenoformans* and *T. carboxydivorans* CODH-ECH gene clusters.

	***C. hydrogenoformans* genome**	***T. carboxydivorans* genome**	***C. hydrogenoformans* CODH–ECH cluster**	***T. carboxydivorans* CODH–ECH cluster**	***E. coli* K12 genome**
C. hydrogenoformans genome	0	*0.302*	*0.023*	*0.048*	*0.155*
T. carboxydivorans genome	**0.143**	0	*0.202*	*0.217*	*0.098*
C. hydrogenoformans CODH–ECH cluster	**0.075**	**0.154**	0	*0.016*	*0.115*
T. carboxydivorans CODH–ECH cluster	**0.088**	**0.133**	**0.036**	0	*0.116*
E. coli K12 genome	**0.170**	**0.153**	**0.236**	**0.212**	0

Figure 19. Heuristic predicton of HGT. By comparing donor (*C. hydrogenoformans*), acceptor (*T. carboxydivorans*) and average genome (*E. coli*) data. Note that full genomes and the target segment (CODH-ECH cluster) sequences are compared in an all vs. all fashion. The comparison is carried out on tetranucleotides (upper half-matrix) and trinucleotides (lower half-matrix). [87]

2. Scope and methods

“Of all my seeking this is all my gain:
No agony of any mortal brain
Shall wrest the secret of the life of man;
The Search has taught me that the Search is vain.” -
Omar Khayyam

2.1 Scope

The process of automated genome annotation, discussed extensively in the Introduction chapter, is becoming ever more important as high-throughput genome sequencing technologies are advancing at an increasing pace. In this regard, many approaches have been proposed and introduced, subsystem based annotation being one of them.

This study concentrates on developing approaches and tools for carrying out extensive subsystem based analysis and annotation of genes, using various QS systems as study cases. In particular, I studied Acyl-Homocerine-Lactone (AHL) based QS systems in *Burkholderia*, *Pseudomonas* and other *Proteobacteria* as well as ComQXPA based systems in *Bacillus subtilis*.

The analysis includes identification of local gene arrangements, gene overlap patterns and studying the potential role of horizontal gene transfer for which we tested a number different methods.

2.2 Data sources and types

This study uses complete genomes, draft genomes and RefSeq entries as data. (Obtained from ftp://ftp.ncbi.nlm.nih.gov/genomes/) At the time of carrying out final analysis there were 2620 complete and 6970 draft genomes.

The main homology detector used throughout this study is HMMER developed by Eddy et.al. [27]. For building homolog recognizers, we used annotated QS genes, mined out from literature. The main proteins that constitute our target subsystem are:

- AHL QS system genes: *luxR, luxI, rsaL, rsaM* (Table 3)

LuxI homologues
YP_002649215.1, YP_003261728.1, YP_003262850.1, YP_003296640.1, YP_003331715.1, YP_003608088.1, YP_004230809.1, YP_003366470.1, YP_004012993.1, YP_004106681.1, YP_004106954.1, YP_004108425.1, YP_003910269.1, YP_003520250.1, YP_003530770.1, YP_003538486.1, YP_003546445.1, YP_003558209.1, YP_003566926.1, YP_003568278.1, YP_003576501.1, YP_003729883.1, YP_003930460.1, YP_003734012.1, YP_003749682.1, YP_003750860.1, YP_003744153.1, YP_003740503.1, YP_003740954.1, YP_003847234.1, YP_003885141.1, YP_004088230.1, YP_004115279.1, NP_521405.1, NP_522340.1, NP_767703.1, NP_106262.1, NP_106661.1, NP_109412.1, NP_385945.1, YP_002965845.1, YP_002966879.1, YP_002346031.1, YP_002347420.1, YP_002426405.1, YP_428477.1, YP_105963.1, YP_106161.1, YP_110894.1, YP_111576.1, YP_001005892.1, YP_554693.1, YP_555669.1, YP_165635.1, YP_167511.1, NP_669050.1, NP_670673.1, YP_528965.1, YP_234707.1, NP_250123.1, NP_252166.1, YP_002232872.1, YP_002234481.1, YP_768958.1, YP_048233.1, NP_793636.1, YP_789671.1, YP_791820.1, NP_903761.1, NP_993604.1, NP_994737.1, YP_674865.1, YP_371808.1, YP_001114940.1, YP_001117676.1, YP_439001.1, YP_439708.1, YP_273860.1, YP_508562.1, YP_071011.1, YP_071751.1, YP_206882.1, YP_914595.1, YP_002551489.1, YP_002549360.1, YP_002541324.1, YP_659946.1, YP_317245.1, YP_776005.1, YP_617566.1, YP_617628.1, YP_838353.1, YP_623506.1, YP_470411.1, YP_473057.1, YP_001024425.1, YP_001025818.1, YP_001077901.1, YP_001078152.1, YP_989942.1, YP_001062290.1, YP_001063210.1, YP_335777.1, YP_337633.1, YP_972130.1, YP_484039.1, YP_486927.1, YP_567542.1, YP_569311.1, YP_530592.1, YP_531903.1, YP_781244.1, YP_001231849.1, YP_001604809.1, YP_001606209.1, YP_001399709.1, YP_001400525.1, YP_001220569.1, YP_001241094.1, YP_001242901.1, YP_001075256.1, YP_001076162.1, YP_002537871.1, YP_001327237.1, YP_453964.1, YP_681952.1, YP_650194.1, YP_651865.1, YP_647981.1, YP_649109.1, YP_002220095.1, YP_855089.1, YP_001161918.1, YP_001163229.1, YP_001347034.1, YP_001349251.1, YP_001143471.1, YP_001583944.1, YP_001860597.1, YP_001811255.1, YP_001531662.1, YP_001534185.1, YP_001761364.1, YP_001476305.1, YP_001888022.1, YP_001893789.1, YP_001083198.1, YP_001844795.1, YP_001777918.1, YP_001779189.1, YP_001641952.1, YP_001772211.1, YP_001758390.1, YP_001776814.1, YP_001783295.1, YP_002158590.1, YP_001948920.1, YP_002423669.1, YP_001927659.1, YP_001203094.1, YP_002128524.1, YP_002976728.1, YP_001989358.1, YP_001991324.1, YP_002282165.1, YP_002495630.1, YP_002496260.1, YP_002497058.1, YP_001906897.1, YP_001908005.1, YP_001832057.1, YP_002826208.1, YP_002317565.1, YP_001979200.1, YP_001985290.1, YP_001719546.1, YP_001720402.1, YP_001873009.1, YP_001873806.1, YP_002265246.1, YP_002327281.1, YP_002439140.1, YP_002441565.1, YP_002923740.1, YP_003019698.1, YP_003002473.1, YP_002955226.1, YP_002909043.1, YP_002934276.1, YP_002360442.1, YP_002947663.1, YP_003964946.1, YP_003941574.1, YP_001603070.1, YP_003070966.1, YP_001715479.1, YP_004144716.1, YP_004145051.1, NP_945673.1
LuxR homologues
YP_002649216.1, YP_003261727.1, YP_003262848.1, YP_003296639.1, YP_003331714.1, YP_003608086.1, YP_004230807.1, YP_003366469.1, YP_004012994.1, YP_004106680.1, YP_004106955.1, YP_004108424.1, YP_003910271.1, YP_003520251.1, YP_003530769.1, YP_003538485.1, YP_003546444.1, YP_003558208.1, YP_003566925.1, YP_003568279.1, YP_003576500.1, YP_003729882.1, YP_003930459.1, YP_003734010.1, YP_003749681.1, YP_003750859.1, YP_003744152.1, YP_003740504.1, YP_003740953.1, YP_003847232.1, YP_003885142.1, YP_004088229.1, YP_004115278.1, NP_521406.1, NP_522339.1, NP_767702.1,

NP_106261.1, NP_106660.1, NP_109411.1, NP_385944.1, YP_002965846.1, YP_002966880.1, YP_002346032.1, YP_002347421.1, YP_002426403.1, YP_428476.1, YP_105961.1, YP_106160.1, YP_110896.1, YP_111575.1, YP_001005891.1, YP_554691.1, YP_555670.1, YP_165634.1, YP_167510.1, NP_669049.1, NP_670674.1, YP_528967.1, YP_234708.1, NP_250121.1, NP_252167.1, YP_002232873.1, YP_002234479.1, YP_768957.1, YP_048234.1, NP_793635.1, YP_789670.1, YP_791822.1, NP_903760.1, NP_993605.1, NP_994736.1, YP_674864.1, YP_371810.1, YP_001114942.1, YP_001117674.1, YP_439002.1, YP_439706.1, YP_273861.1, YP_508561.1, YP_071012.1, YP_071752.1, YP_206883.1, YP_914594.1, YP_002551488.1, YP_002549361.1, YP_002541325.1, YP_659944.1, YP_317246.1, YP_776003.1, YP_617565.1, YP_617627.1, YP_838351.1, YP_623508.1, YP_470410.1, YP_473056.1, YP_001024423.1, YP_001025820.1, YP_001077903.1, YP_001078154.1, YP_989940.1, YP_001062292.1, YP_001063209.1, YP_335776.1, YP_337635.1, YP_972129.1, YP_484040.1, YP_486928.1, YP_567541.1, YP_569310.1, YP_530593.1, YP_531902.1, YP_781245.1, YP_001231850.1, YP_001604810.1, YP_001606210.1, YP_001399708.1, YP_001400524.1, YP_001220570.1, YP_001241092.1, YP_001242900.1, YP_001075258.1, YP_001076161.1, YP_002537872.1, YP_001327236.1, YP_453965.1, YP_681951.1, YP_650193.1, YP_651866.1, YP_647982.1, YP_649110.1, YP_002220093.1, YP_855090.1, YP_001161917.1, YP_001163230.1, YP_001347033.1, YP_001349253.1, YP_001143472.1, YP_001583946.1, YP_001860599.1, YP_001811253.1, YP_001531661.1, YP_001534186.1, YP_001761363.1, YP_001476304.1, YP_001888024.1, YP_001893790.1, YP_001083200.1, YP_001844797.1, YP_001777917.1, YP_001779191.1, YP_001641953.1, YP_001772212.1, YP_001758389.1, YP_001776815.1, YP_001783296.1, YP_002158591.1, YP_001948918.1, YP_002423670.1, YP_001927660.1, YP_001203095.1, YP_002128523.1, YP_002976727.1, YP_001989359.1, YP_001991323.1, YP_002282164.1, YP_002495629.1, YP_002496262.1, YP_002497059.1, YP_001906896.1, YP_001908006.1, YP_001832058.1, YP_002826207.1, YP_002317567.1, YP_001979199.1, YP_001985289.1, YP_001719545.1, YP_001720401.1, YP_001873010.1, YP_001873807.1, YP_002265247.1, YP_002327279.1, YP_002439139.1, YP_002441567.1, YP_002923741.1, YP_003019697.1, YP_003002472.1, YP_002955225.1, YP_002909041.1, YP_002934275.1, YP_002360441.1, YP_002947664.1, YP_003964947.1, YP_003941575.1, YP_001603072.1, YP_003070967.1, YP_001715477.1, YP_004144717.1, YP_004145052.1, NP_945674.1,
RsaL homologue
NP_250122.1, YP_001349252.1, YP_001860598.1, YP_001888023.1, YP_002441566.1, YP_002794907.1, YP_003608087.1, YP_003847233.1, YP_003910270.1, YP_554692.1, YP_791821.1
RsaM homologues:
YP_439707.1, YP_001062653.1, YP_776004.1, YP_001117675.1

Table 3. Proteins used for building HMM recognizers for searching respective genes.

2.3 Computational tools and media

During the course of this study, many tools, pipelines and environments have been developed using various programming languages and third party libraries, such as (in decreasing order of involvement): Python, MATLAB, R.

Algorithms which were implemented using Python programming language, were mainly deployed as tools for Galaxy [88] framework, for further ease of use and sharing between peers.

Once the data and results were calculated, they were mostly organized as web pages for demonstration, using various Python web frameworks. The links to results of each chapter will be provided in respective sections.

In depth explanation and analysis of www-related computational tools will be covered in Chapter 6.

2.4 Methods and algorithms

2.4.1 Homology detection

The first step of this study was choosing of tool for remote homology detection. The task was to search for homologs of QS genes, for which had pre-collected library of sequences. As described in Chapter 1.3, we had two approaches to go with:

- Sequence alignment based similarity search
- Statistical profile based similarity search (Hidden Markov Models)

With the sequence alignment based option, we could not go with plain optimal alignments methods (such as Smith-Waterman [17] or Needleman-Wunsch [16] algorithms), since the amount of data that we were intending to scan was too big for running them on. So, instead, we had to choose from sub-optimal heuristic approaches, which run faster. The most popular and robust example of such tool is BLAST [29]. The idea was building BLAST databases for each set of QS genes, and querying each available genome sequence (*fna* files of chromosomes of corresponding genome) against these databases, searching for homologous genes (genome regions with high similarities).

As for HMMs, the most popular and robust implementation is HMMER [24]. In similar fashion with BLAST, we first build statistical profile of sequence groups using our sequence libraries (further as HMM recognizers), and scan input sequences in search for high score hits. One other big difference of HMMER from BLAST is that, HMMER runs only on protein sequences, whereas BLAST can run on both protein and DNA sequences. Therefore, BLAST would allow us to handle draft genomes, where not all the genes are annotated, if it is at least as good as HMMER. HMMER takes as input concatenated fasta files of protein sequences. So, as opposed to BLAST inputs, where each chromosome was represented in terms of one single long sequence, here we feed a fasta file for each chromosome, containing annotated protein sequences in it.

My aim was to run BLAST and HMMER on test genomes using the same libraries, and to assess, which one of the tools performs better. If we run both BLAST and HMMER on protein sequences, HMMER clearly outperforms BLAST, because BLAST gives false positives to the threshold value of e-value = 10^{-10}. What we were interested in is, running BLAST on DNA sequences, and compare the hit regions with correspond to annotated regions and calculate the coverage of two regions.

After running our experiment, we observed that on raw DNA sequences, false positive rate of BLAST is high. And even the hits which corresponded to genes, gave significantly erroneous boundaries. BLAST can compete with HMMER in accuracy, only if it is run on sliced sequences, which are predicted to be genes. In case of this study, GLIMMER [30] was used (Markov Model (MM) based bacterial gene finder) to detect tentative gene regions.

In order to be able to run HMM search on DNA sequences, one needs to translate DNA sequences to protein sequences in all six frames. One of the translations will be hit by HMM, if there is a homologous region (Figure 20). The problem which arises with this approach is that HMM hit rarely covers the entire gene. Instead, outputs a local region (domain) with higher matching score, than longer region with lower matching score. As a result, that domain hit needs to be extended to the actual borders of the gene. If there are more than one domain hits, they need to be merged first, and then extended to global borders of the gene.

Global borders of the gene are the coordinate of nearest methionine (M) at the N-terminal, and any of stop codons at the C-terminal (Figure 21). Stop codons were proven to be reliable for estimation of coordinate of C-terminal. Whereas N-terminal coordinate estimation is uncertain, since it is hard to decide if the Methionine found while spanning towards left is the first codon of gene, without additional information.

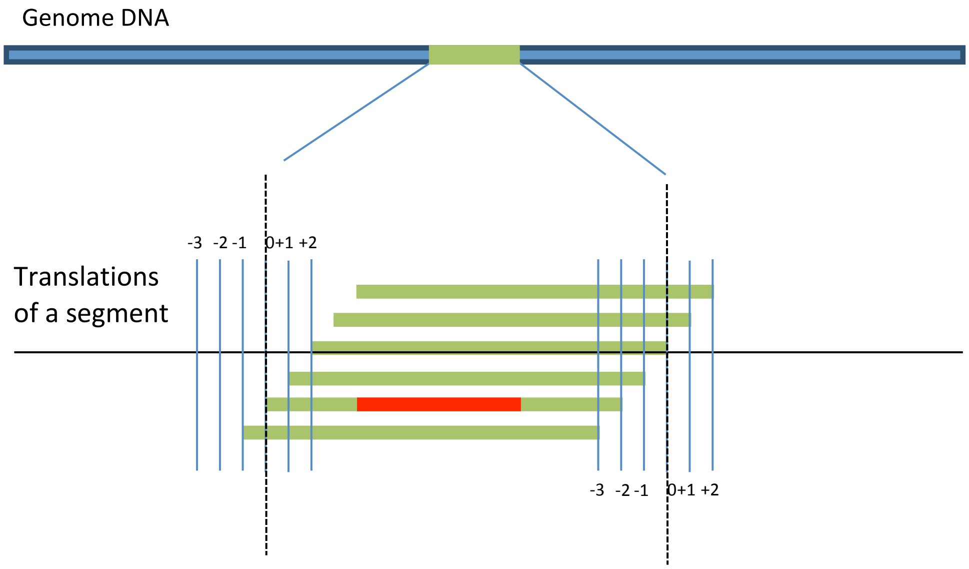


Figure 20. Searching with HMM on translated sequences of DNA segment.

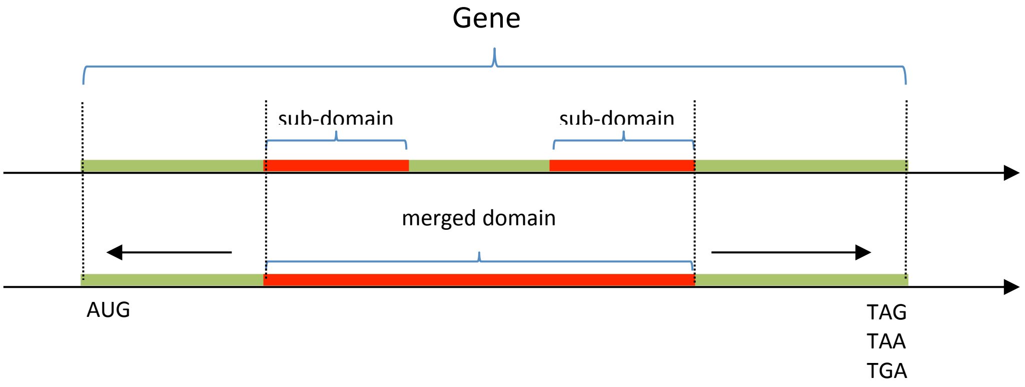


Figure 21. Domain merging. When two domains are found, they get merged into one domain. After that, the borders of super-domain get extended to borders of gene, which are the coordinates of nearest *methionine* (M) and stop codons.

2.4.2 Mining QS systems components and data.

The core algorithm for analyzing QS systems can be summarized with flowchart in Figure 22 .

While different QS systems require processing steps and input data differing from each other, analysis of any kind of QS system requires the steps depicted above. For instance, detecting clusters of genes („topologies”, described in next chapter) of AHL systems is completely different from that of COM systems. Furthermore, analyses of different QS systems lead to different biological interpretations. In order to create a generic template of pipeline for conducting automated analysis of different QS systems, I developed Galaxy framework pipelines. In this way, it was possible to deploy different automated tasks using smaller sub-units of pipeline (called Galaxy tools) in different combinations, and develop individual sub-routines for QS system of interest when necessary.

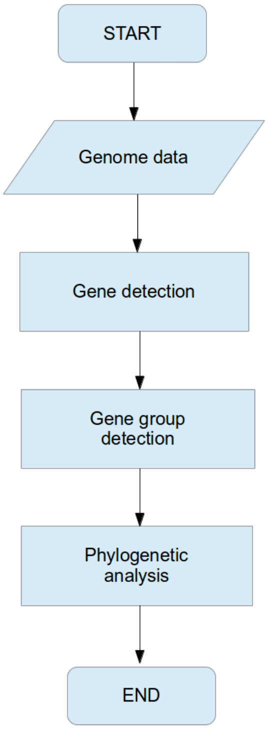


Figure 22. Core algorithm of subsystem based analysis of QS systems.

An example case of workflow of a Galaxy pipeline for analyzing AHL based QS system is depicted in Figure 23

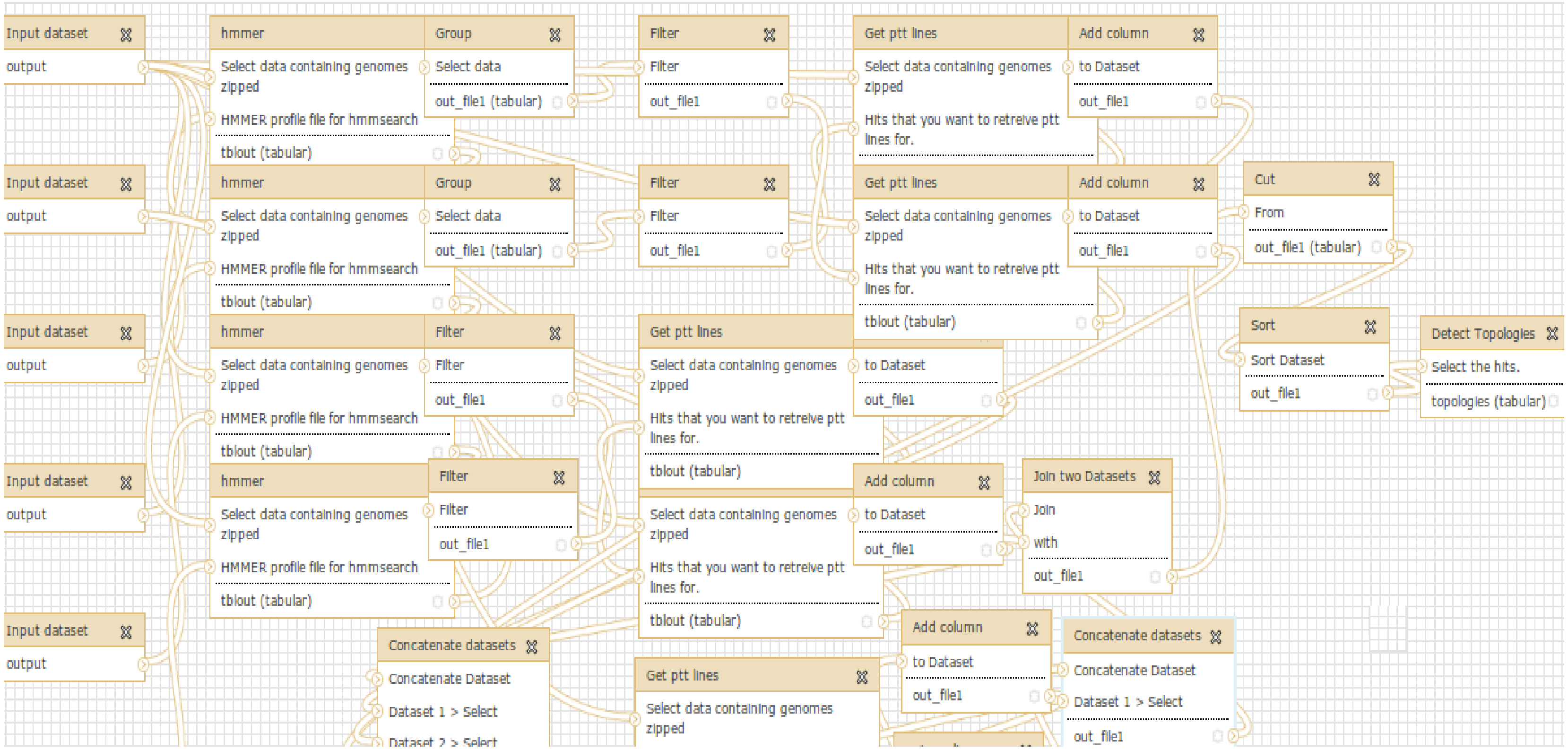


Figure 23. Galaxy workflow for mining and annotating AHL based QS systems. The vertical line of nodes on the left represent input fields. The ultimate result is the rightmost node, output of which is the list of topologies with detailed information about them. While some of the nodes in this picture are specific to AHL based QS system, most of the tools are usable for working with any other kinds homology detection based systems.

3. Results and discussion

"Cogito ergo sum" - René Descartes

3.1. Chromosomal arrangement patterns.

The genes constituting a QS system tend to be located together on the chromosome as discussed in the introduction. Our subsystem based approach mainly focuses on genes adjacent or located close to each other. Throughout this study, group of adjacently located QS genes will be called as 'topologies'. The topologies were selected with the threshold of maximum 3000 base pairs of distance from each other. In other words, if genes are more than 3000 base pairs apart from each other, they won't be considered as a single topology even if they are located one after another, without any genic regions in between.

For simplicity in depiction and formulation of topologies, *luxI, rsaL, rsaM* and *luxR* genes are represented as I,L,M and R respectively. And for representing strand of a gene on which the gene is located, an arrow over letter is used. While genes can be located on either positive or negative strands, the arrows only emphasize the direction of genes with respect to each other. Therefore, an arrow pointing towards right side can mean either negative or positive strand. So, as a result a notion like $\overrightarrow{R}\overleftarrow{L}\overrightarrow{I}$ means that the genes might be located in order of both *luxR-rsaL-luxI* and *luxI-rsaL-luxR*. What it emphasizes is the order of occurrence of genes and the fact that *luxR* and *luxI* are located on the same strand, whereas *rsaM* is located on the opposite strand.

The questions targeted with above described topologic approach were:

1) How well the topologies are conserved?
2) Do similar topologies group together in phylogenetic trees of organisms which carry them?

The process of generating annotation and topology detection were implemented in terms of Galaxy toolboxes as described in previous chapter.

Apart from cluster forming genes, some of genes which are involved in QS machinery, can be found far away from each other. One particular example of this behavior is that of *luxR* [89], which is found on genomes as a part of a neighborhood of related *luxI*, *rsaL* and *rsaM* genes, as well as surrounded by genes which are not involved in QS system (those *luxR* genes which are not located together with other QS genes will be named as solo R genes, and are extensively studied later on in this chapter). To carry out analysis of locations of these seemingly distant gene locations for any possible common patterns, I created circular diagrams of chromosomes with QS genes and topologies projected on circular diagrams according to their location coordinates.

The aim in building these circular diagrams was to understand if relative locations of QS components are conserved with respect to each other as well as with respect to the origin of replication.

The origin of replication is located where GC skew of chromosome is at minimum level. GC skew is a metric calculated over genome/chromosome using non-overlapping sliding windows. For each considered window, occurrence times of C and G nucleotides are counted, and skew value is set as shown in (Eq. 9)

$$\text{GC skew} = \frac{\text{n(G)-n(C)}}{\text{n(G)+n(C)}} \qquad (9)$$

For locating origin of replication, one calculates cumulative sum of GC skew values as described above, and finds global minimum over the range of values (Figure 24). The size of window for calculating above mentioned metrics is very important if location of origin of replication is subject to precise measurements. But in this case, we only need to project the cartesian coordinate to a polar coordinate, therefore, commonly accepted window size of 100 was used.

As a result, we have the following entities on the circular diagrams: beginning of coordinate reference (coordinate 0, from where we start counting the position), estimated origin of replication and QS system topologies or isolated genes. Figure 25 shows that there is one particular RLI topology and one isolated R gene on the chromosome NC_018672 of *Burkholderia phenoliruptix*.

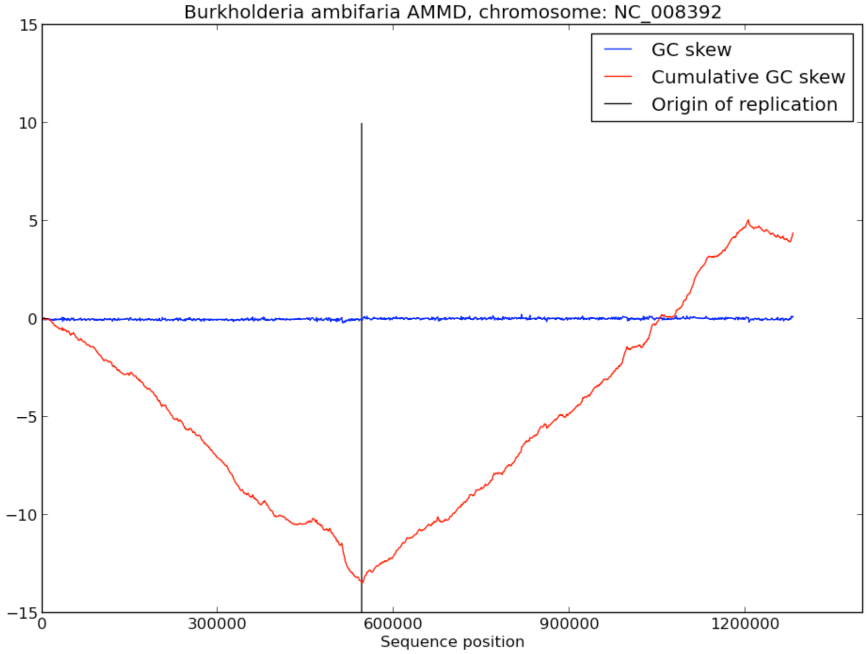


Figure 24. GC skew. Also cumulative GC skew and supposed *origin of replication* of chromosome *NC_008392* of bacteria *Burkholderia ambifaria AMMD*.

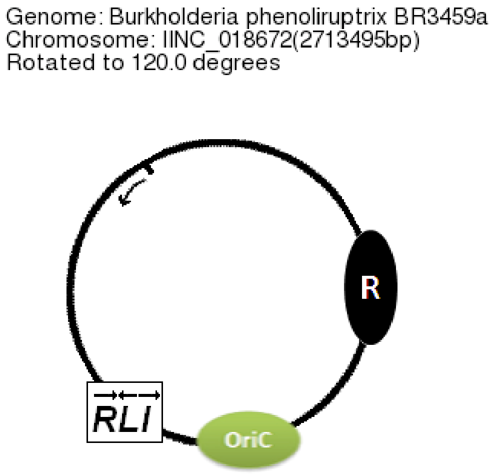
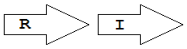
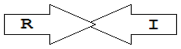
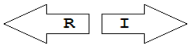
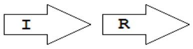
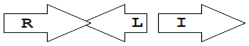

Genome: Burkholderia phenoliruptrix BR3459a
Chromosome: IINC_018672(2713495bp)
Rotated to 120.0 degrees
R
RLI
OriC

Figure 25. Circular diagram. Chromosomal arrangements of AHL based QS system genes, origin of replication and reference points.

3.2. Local gene arrangement patterns

Based on the definitions and assumptions made in Chapter 3.1, all possible combinations of AHL based QS system genes were searched for. The search was carried out in complete and draft genomes. The topological combinations found in bacterial genomes are summarized in Table 4. Each combination is given a code name (Field: ID) for further convenience.

The classification scheme of topologies used in this study was largely taken from a study by Gelencser et.al [90], according to which we classify the topologies either as *simple topologies* wherein we have at least one *luxR* and one *luxI* genes involved with 1-2 intervening genes, and *complex topologies* wherein the topological patterns can be irregular and we can have more intervening genes. While the genes of interest are only R,I,M or L genes, sometimes there are genes which were not recognized as one of the mentioned genes, yet are located inside a topology. For this kind of cases, we use X to denote these non-QS genes. The X genes' strand information was not taken into account. To be able to handle the cases when there are variable number of X genes are present in topology, the regular-expression like notation was used. So, for instance $\vec{I}X(>7)\vec{R}$ means that there are more than 7 X genes between I and R genes.

ID	Pattern	Gene topology
Simple Topologies		
R1	$\vec{R}\vec{I}$	
R2	$\vec{R}\overleftarrow{I}$	
R3	$\overleftarrow{R}\vec{I}$	
R4	$\vec{I}\vec{R}$	
L1	$\vec{R}\overleftarrow{L}\vec{I}$	

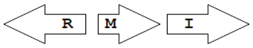
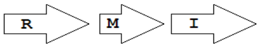
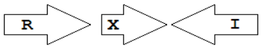
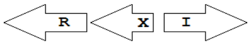
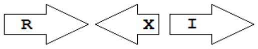
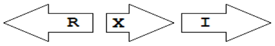
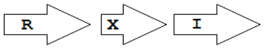
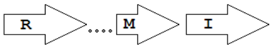
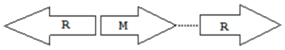
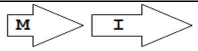
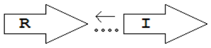
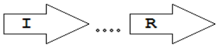
M1	$\overleftarrow{R}\overrightarrow{M}\overrightarrow{I}$	
M2	$\overrightarrow{R}\overrightarrow{M}\overrightarrow{I}$	
X1	$\overrightarrow{R}\overrightarrow{X}\overleftarrow{I}$	
X2	$\overleftarrow{R}\overleftarrow{X}\overrightarrow{I}$	
X3	$\overrightarrow{R}\overleftarrow{X}\overrightarrow{I}$	
X4	$\overleftarrow{R}\overrightarrow{X}\overrightarrow{I}$	
X5	$\overrightarrow{R}\overrightarrow{X}\overrightarrow{I}$	
Complex Topologies		
M3	$\overrightarrow{R}X(2-11)\overrightarrow{M}\overrightarrow{I}$	
M4	$\overleftarrow{R}\overrightarrow{M}X(<7)\overrightarrow{I}$	
M31	$\overrightarrow{M}\overrightarrow{I}$	
X6	$\overrightarrow{R}\overleftarrow{X}(7)\overrightarrow{I}$	
X7	$\overrightarrow{I}X(>7)\overrightarrow{R}$	

Table 4. Topological arrangements of AHL QS system genes.
Listed topologies were found in bacterial genomes.

Taking into consideration the strand information of genes leads to definition of different sorts of arrangements:

- *Convergent genes*: C-termini of genes face each other and are located on opposite strands. E.g. R2, X1.
- *Divergent genes:* N-termini of genes face each other and are located on opposite strands. E.g. R3, M1, X2
- *Tandem genes*: The genes are located on the same strand and transcribed in the same direction. E.g. R1, R4, M2

ID	*Burkholderia*	*Pseudomonas*	All
R1	48	96	538
R2	2	48	502
R3	0	4	78
R4	0	0	20
M1	213	3	585
M2	0	0	3
M3	198	0	198
M31	20	0	46
X2	0	0	9
X3	75	0	93
X4	0	0	18
X5	6	0	9
X6	5	0	5
X7	0	0	5
L1	36	81	123
M4	4	0	4

Table 5. Statistics of topologies for *Burkholderia* and *Pseudomonas*.

Table 5 provides information on number of times a particular topology was observed in *Burkholderia*, *Pseudomonas* and in general. It is evident from the table that mainly the AHL based QS system is found in *Burkholderia*.

Once topologies were assigned and annotated for each found case, distribution of topologies among phylogenic clades was analyzed. For this purpose, phylogenic trees from *luxI* genes were created for *Burkholderia* and *Pseudomonas* genera. Topologies turned out to be distributed more coherently within the phylogenetic tree than taxonomies of organisms carrying them.

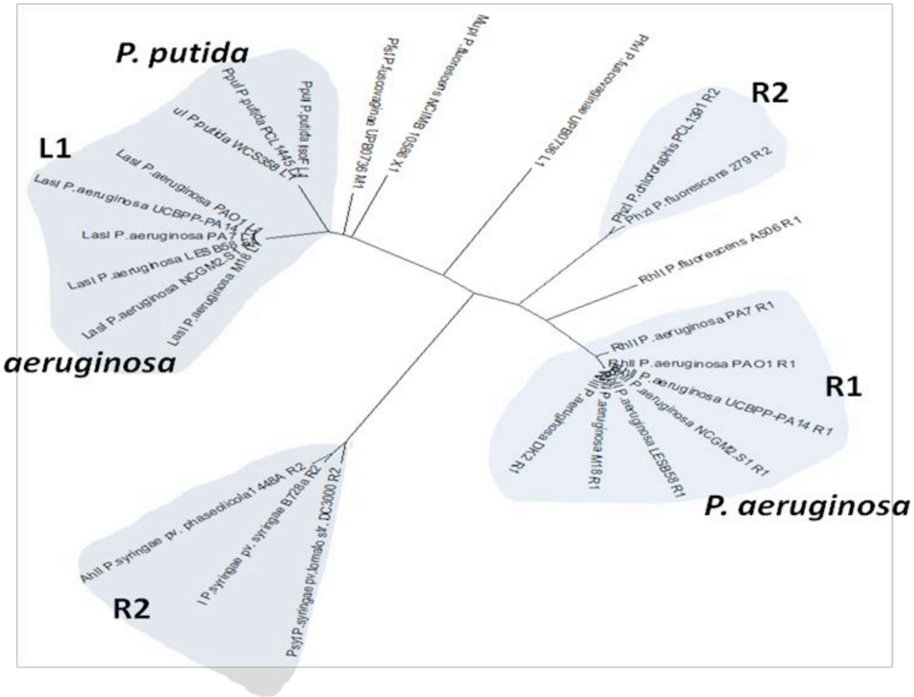

P. putida
L1
R2
R1
P. aeruginosa
P. aeruginosa
R2
PpuI P.putida IsoF L1
PpuI P.putida PCL1445 L1
uI P.putida WCS358 L1
LasI P.aeruginosa PAO1 L1
LasI P.aeruginosa UCBPP-PA14 L1
LasI P.aeruginosa PA7 L1
LasI P.aeruginosa LES B58 L1
LasI P.aeruginosa NCGM2.S1 L1
LasI P.aeruginosa M18 L1
PsI P.fuscovaginae UPB0736 M1
MupI P.fluorescens NCIMB 10586 X1
PfvI P.fuscovaginae UPB0736 L1
PhzI P.chlororaphis PCL1391 R2
PhzI P.fluorescens 2-79 R2
RhlI P.fluorescens A506 R1
RhlI P.aeruginosa PA7 R1
RhlI P.aeruginosa PAO1 R1
RhlI P.aeruginosa UCBPP-PA14 R1
RhlI P.aeruginosa NCGM2.S1 R1
RhlI P.aeruginosa LESB58 R1
RhlI P.aeruginosa M18 R1
RhlI P.aeruginosa DK2 R1
AhlI P.syringae pv. phaseolicola1448A R2
I P.syringae pv. syringae B728a R2
PsyI P.syringae pv.tomato str. DC3000 R2

Figure 26. Phylogenic tree of *Pseudomonas* having *luxI* genes.

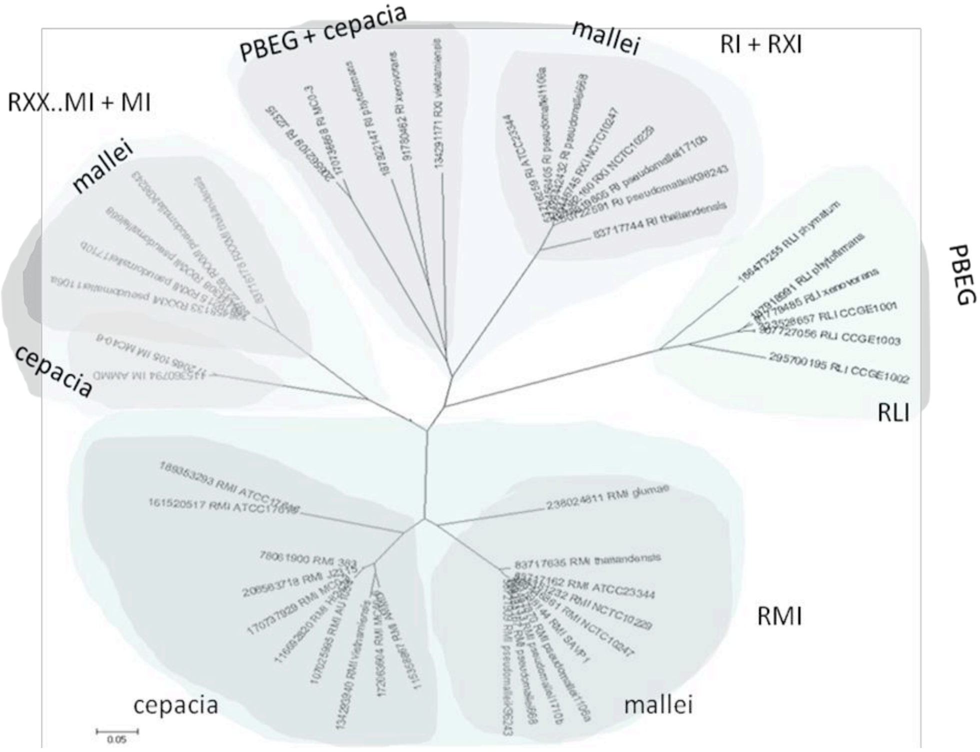


Figure 27. Phylogenic tree of *Burkholderia* having *luxI* genes.

Figure 26 shows that *Pseudomonas* do not group according to taxonomic distribution. For instance, *Pseudomonas aerueginosa* species were split into two distant clades, whereas every clade is represented by one or more topology types. Similar interpretation is valid for *Burkholderia* as well. Figure 27 shows that grouping according to topologies is more coherent than that of taxonomy.

3.3. Gene overlap patterns

Overlapping genes are the genes, coding regions of which collide with each other. It is believed that gene overlaps arise to make the length of genome shorter thus minimizing the cost of maintenance [91] . The overlaps arise as a result of point mutation in either 5' or 3' ends of genes. According to the strands of overlapping genes, the gene overlaps can be categorized as in Figure 28

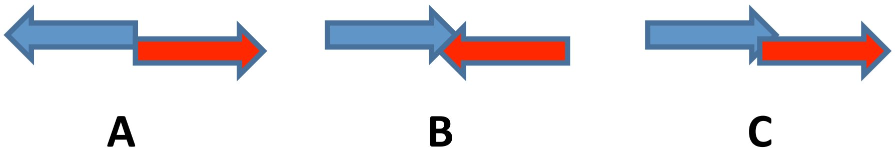


Figure 28. Types of overlaps I. Overlapping genes can be divergent (A), convergent (B) and unidirectional (C)

According to the locations of overlapping genes, gene overlaps can be categorized as Figure 29. While the overlaps touching each other at end points (Figure 29, D) by definition are not overlaps, but they are usually surveyed together with gene overlaps.

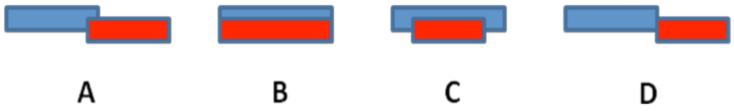


Figure 29. Types of overlaps II. Overlapping genes can be forms of: terminal overlap (A), equal or almost equal (B), one containing another (C) and touching each other at end points (D).

Mathematically, two overlapping genes can be defined as:

$$(x_1y_1, x_2y_2) = \{x_1y_1, x_2y_2 \in Z, \Sigma \mid (x_1 - y_2) \times (x_2 - y_1) \geq 0\}$$

where Z : integer number, Σ: range of gene coordinates, ($x_i y_i$) is coordinate of a gene. The condition $(x_1 - y_2) \times (x_2 - y_1) = 0$ holds when genes have terminal overlap (Figure 29, A).

A script was wrapped as a Galaxy tool for scanning all consecutive AHL QS genes and parsing out those which match the above mentioned rule.

The gene overlap cases among R,I,M and L genes are summarized in

Table 6.

Topology	Overlapping genes	Occurrence
RI	*luxR, luxI*	217
RLI	*luxR, rsaL*	40
RXI	X, *luxI*	3
RXI	X, *luxR*	3
RXMI	X, *rsaM*	3

Table 6. Found cases of gene overlaps for AHL QS system genes.

3.4. Patterns of horizontal gene transfer

Quorum sensing is governed by small sets of core genes that govern a number, sometimes a large number of genes within bacteria. It is a conspicuous fact that there are genera and families where only a few members carry known QS genes, so one might hypothesize that those few members acquired the QS property by horizontal gene transfer (HGT). Also, many bacterial species carry solo *luxR* genes [55] that may be responding either to QS signals produced by a nonadjacent signal synthase gene, or, they may respond to external signals. The evolutionary fate of these genes is not well understood. There are no explicit data on the HGT properties of these genes, and since our survey indicated a large number of novel QS genes, it looked a plausible step to check if they can be linked to potential HGT events. Since we have several thousand of genes to check, the first step was to develop a computational method that can check the properties of these genes on an equal footing.

3.4.1 Testing various vector descriptions and comparison methods

Various methods proposed for HGT detection are based on vector representations (di-, tri- and tetranucleotide etc.). Tetranucleotides are currently used the most, but our initial step was to make a systematic test on the over 9000 genomes that are presently in the databases.

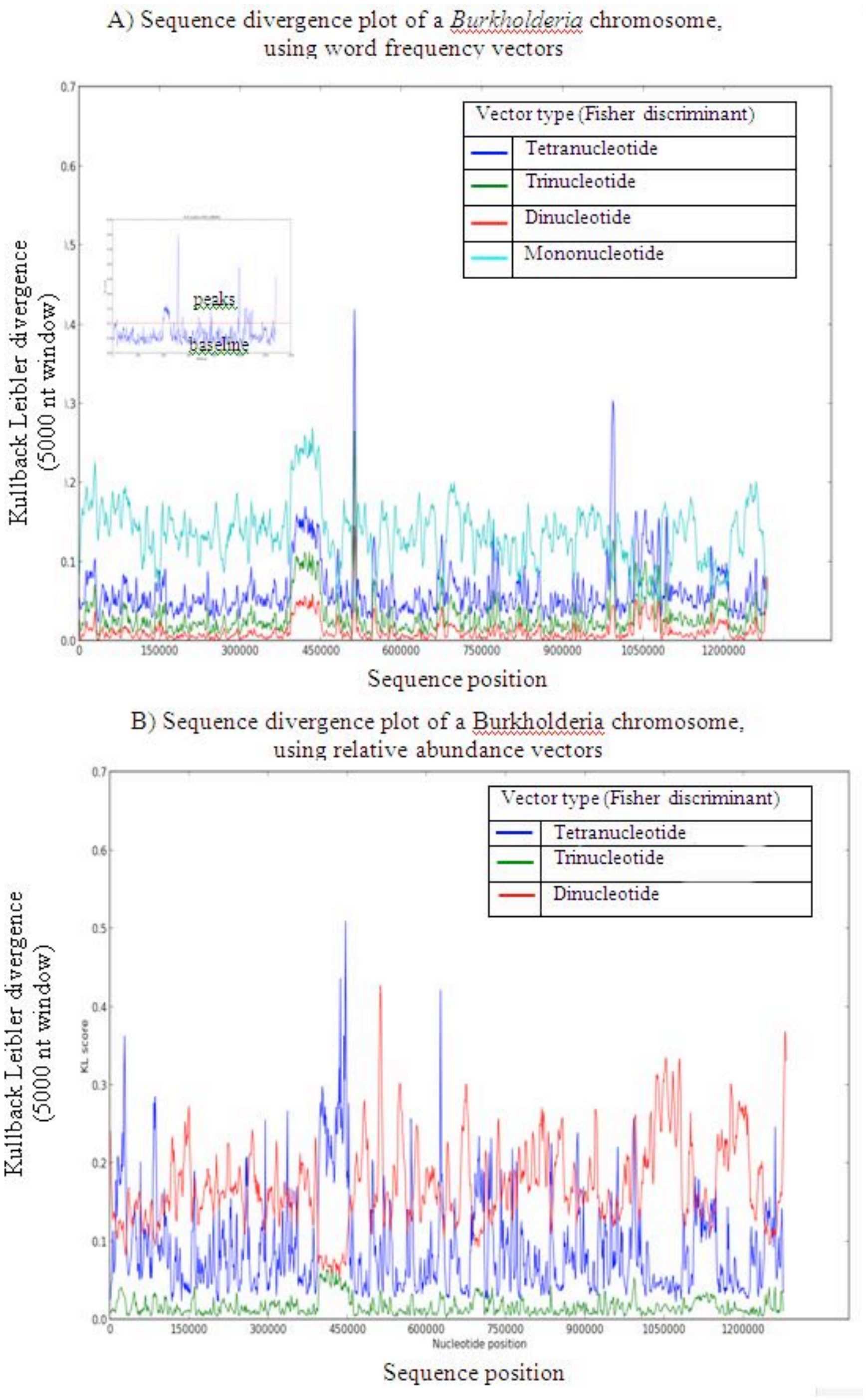
Figure 30 shows a qualitative comparison of different vector representations. A window of 5000 nucleotides length was slid along the chromosome sequence, and a local vector description was compared to the vector of the entire chromosome as described in the introduction (Equation 8). The *y* axis is the Kullback-Leibler divergence which is a measure of the local difference, i.e. it is high for regions that may have arrived by HGT to the genome. It is apparent that the chromosome contains two conspicuous regions, a plateau-like region around positions (400000, 450000) and a sharp peak at around (508000, 515000).

Figure 30 also contains a numerical performance measure for the vector type. The Fisher's discriminant metric in context of Linear Discriminant Analysis is a widely used measure for comparing inter-group variations with within-group variations [92]. Relying on Fisher's discriminant, we devised a separation metric which suits our task (Equation 10). We defined the groups by first dividing the values to "peak" and "baseline" (Figure 30 A, inset) and calculated the separation between these two groups from the average and standard deviation values of within-group and between-group comparison of the vectors:

$$F = \frac{(y_{within} - y_{between})^2}{(sd_{within}^2 + sd_{between}^2)} \quad (10)$$

This is a standard approximation that in our case measures the performance of a vector type to separate peaks from the baseline.

Figure 30 shows that the performance of the vector types is different, noisy plots are associated with low discriminant values, which is an indicator of poor performance. The relative abundance plots are noisy and have low discriminant values. The simple frequency vectors are less noisy and have higher discriminant values. The mononucleotide plot seems to be the noisiest, but interestingly, the best discriminant value is seen at trinucleotide plots, and not at tetranucleotides, as expected. One can suspect that the window of 5000 may be too short so that the tetranucleotide vectors are too sparse.



A) Sequence divergence plot of a *Burkholderia* chromosome, using word frequency vectors

B) Sequence divergence plot of a Burkholderia chromosome, using relative abundance vectors

Figure 30. Word frequency plot of a *Burkholderia* chromosome (NC_008392).

The comparison in Figure 30 confirmed that there are differences between the vector types, but this was calculated only for a single chromosome. So we set up a more systematic evaluation where we compared about 2700 complete genomes. As there are no standard methods for this evaluation, first we tested the sensitivity of the method on a qualitative basis.

The principle shown in Figure 31 – is to represent genomes as vectors and then to compare them in an all-vs-all fashion. In the figure we see the comparison of three genera, *Burkholderia*, *Escherichia* and *Chlamydia*. The figure clearly shows that the method can well distinguish genome groups from each other; within genus vectors are clearly lower than between genera vectors. The overall separation between groups, calculated as the discriminant is 2.038 in this case. The figure also illustrates an important property of this approach: One can calculate an all-vs-all comparison matrix for any vector type, and one can use the matrix to calculate species separation, genus separation etc., for any taxonomic level. The heat map in Figure 31 shows that the genera separate well from each other, but there is less separation within the genera.

The comparison of all 2771 bacterial genomes is shown in Figure 32. The picture is less clear-cut. The discriminant value at the species level is 0.576.

With these preliminaries I compared 4 different vector descriptions: mono-, di-, tri and tetranucleotides, in 3 different representations: simple frequency vectors, relative abundance vectors and structural equivalence vectors (described in Figure 16). The comparison was carried out at species, genus and family levels using the discriminant value as a performance measure. (Table 7)

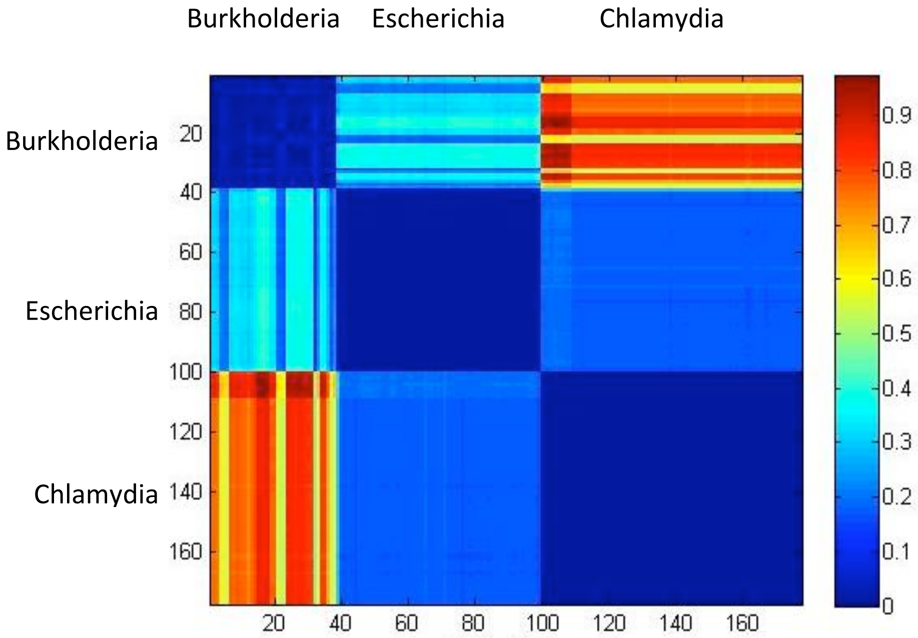


Figure 31. The comparison of 3 genera in terms of Kullback-Leibler. Divergence is calculated between tetranucleotide frequency vectors. Note that the diagonal matrices are the within group comparisons, the off diagonal matrices are the between group comparisons.

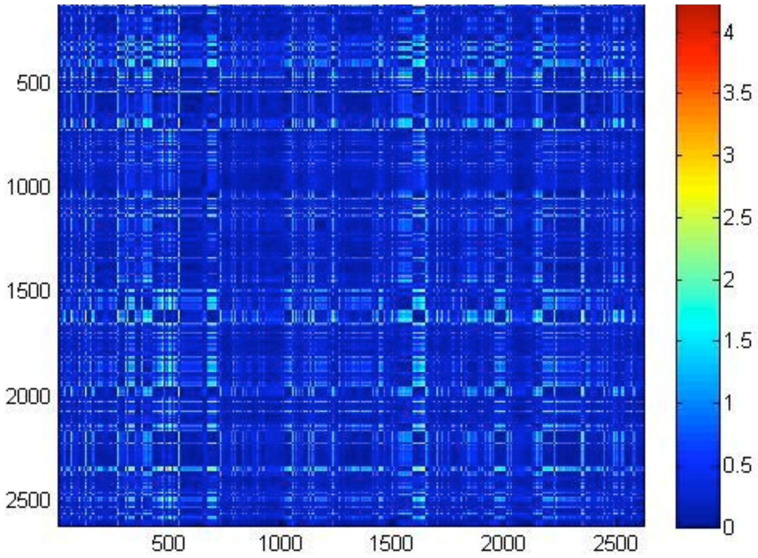


Figure 32. All-vs-all comparison.

The comparison is made for all 2771 bacterial genomes in terms of Kullback-Leibler divergence calculated between tetranucleotide frequency vectors. The discriminant value calculated for species separation is 0.5729.

Species level

	Frequency	SE-frequency	Rel. abundance	SE-Rel. abundance
Mononucleotide	0.7053	0.7035	NA	NA
Dinucleotide	0.897	0.8588	1.5395	1.1667
Trinucleotide	1.1107	1.0281	2.192	1.8541
Tetranucleotide	1.3684	1.2276	2.6753	2.2309

Genus level

	Frequency	SE-frequency	Rel. abundance	SE-Rel. abundance
Mononucleotide	0.6394	0.6382	NA	NA
Dinucleotide	0.7946	0.7665	1.1132	0.8142
Trinucleotide	0.9591	0.9005	1.4621	1.1835
Tetranucleotide	1.1462	0.9733	1.6768	1.3417

Family level

	Frequency	SE-frequency	Rel. abundance	SE-Rel. abundance
Mononucleotide	0.3552	0.3544	NA	NA
Dinucleotide	0.4486	0.4276	0.8767	0.6034
Trinucleotide	0.5384	0.4999	1.1423	0.8982
Tetranucleotide	0.6279	0.4995	1.2662	0.986

Table 7. Evaluation of vector descriptions for various word sizes.

3.4.2 Prediction of HGT in QS genes

Based on the preliminary evaluation described in the previous paragraph, we made a wholesale comparison of all QS genes, both for those that occur in QS topologies, and those that are *luxR* solos. We took +/- 5000 neighborhood of the QS genes as a "unit", calculated tetranucleotide divergence and GC divergence for them and plotted them as a two-dimensional scatter plot. The scatter plot shows that QS genes are not especially prone to HGT, since their divergence values do not substantially differ from the genomic averages.

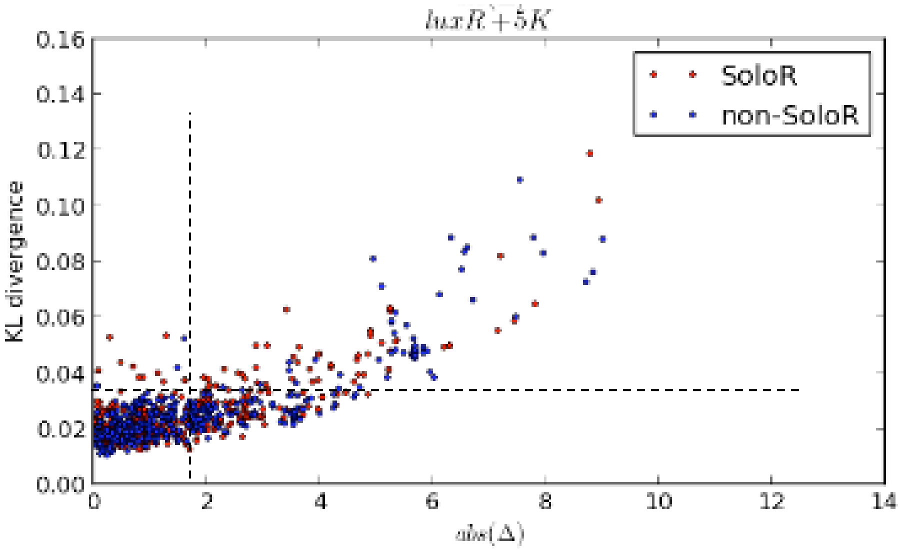


Figure 33. A scatter plot of QS genes in various genomes. A total of 3464 segments, corresponding to 1081 QS topologies and 2383 solo *luxR* proteins were plotted. The dotted lines indicate the values of genomic averages: KL=0,0293, abs(Δ) = 1,79.

It is conspicuous that topology-bound and solo *luxR* genes do not separate in the plot, which means that neither type is more prone to HGT than the other type. Also, they occur roughly equally in the outliers' region. In this region we find genes from the following genera: *Pantoea, Burkholderia, Pseudomonas, Polymorphum, Gluconacetobacter, Rhodopseudomonas, Rahnella, Halothiobacillus, Gluconacetobacter, Micavibrio.*

In most of these genera, the species harboring the QS-related gene is one of very few within the genus, so the outlier property is correlated with the unusual taxonomic distribution. This means that in these cases we can expect HGT to play a role at least in principle.

Another approach to test the HGT is to taxatively check the inequalities listed in the introduction. This can be numerically checked for all QS genes by pre-calculating all (gene-to-gene, gene-to genome, genome-to-genome) comparison values in the form of a distance matrix and checking the inequalities for all genes. This requires minutious manual checking, but it is based on a simple principle: A *luxR* gene should be substantially closer to a foreign genome, than to its own "host" genome. Substantially is mean in a qualitative sense, Techtmann et al [87] used a criterion that the average genome was roughly as distant from the segment in question, as the host genome itself. To check this principle I listed the 10 most conspicuous luxR gene neighborhoods.

LuxR gene [2], strain, chromosome id/Topology	KL Distance[1]		
	Own genome	Nearest genome with QS genes	Average genome
YP_195352, Aromatoleum_aromaticum_EbN1_uid58231, NC_006823, R	0.00963872	0.000895	0.03339
YP_486928, Rhodopseudomonas_palustris_HaA2_uid58439, NC_007778, RI	0.00951471	0.002026	0.02361
YP_005200648,Rahnella_aquatilis_CIP_78_65___ATCC_33071_uid86855, NC_016818, RI	0.00918153	0.002333	0.014492
YP_003262848, Halothiobacillus_neapolitanus_c2_uid41317, NC_013422, RMI	0.00860706	0.002003	0.025017
YP_001603072, Gluconacetobacter_diazotrophicus_PAl_5_uid61587, NC_010125, RXI	0.00811162	0.002464	0.024895
YP_004117160, Pantoea_At_9b_uid55845, NC_014837, R	0.00790213	0.001223	0.014128
YP_776923, Burkholderia_ambifaria_AMMD_uid58303, NC_008391, R	0.00742393	0.001228	0.014138
YP_004864636, Micavibrio_aeruginosavorus_ARL_13_uid73585, NC_016026, RR	0.00682175	0.001165	0.029306
YP_008258288,Salmonella_enterica_serovar_Bareilly_CFSAN000189_uid212971, NC_021817, R	0.00668614	0.001419	0.023462
YP_006325991, Pseudomonas_fluorescens_A506_uid165185, NC_017911, R	0.0062857	0.001402	0.012823

Table 8. KL values for 10 loci of luxR subject to possible HGT event. 1) Symmetrized Kullback-Leibler divergence. 2) With 5000 nucleotides flanking on both ends. 3) Average distance of the *luxR* neighborhood from all genomes.

Table 8 shows that the LuxR neighborhoods are not convincingly far away from their own genomes, i.e. the main condition of HGT is generally not fulfilled. The only species where this condition is fulfilled is *Aromatoleum aromaticum* which is substantially nearer to a *Pseudovibrio GE062*, as species which has two solo *luxR* genes. *Aromatoleum aromaticum* also has only solo R genes, so there may be a chance that there was HGT between these two organisms, however there is no proof the solo R genes are involved in QS. Based on this we conclude that HGT between distant species is not likely to play a major and general role in the evolution of QS *luxR* genes. The term “distant species” is emphasized here because the tetranucleotide signature method tested here is not sensitive to the small differences that may exist between closely related species. So our analysis does not entirely rule out that *luxR* regulated clusters may be exchanged between closely related species but our analysis is not able to detect such transfers. The lack of HGT in AHL systems was also suggested by an earlier publication [93] , but that conclusion was based on a study of a few *Vibrio* species. Our study is the first comparison that included a large number of genomes, and the conclusion seems to confirm this earlier study.

3.5. Biological applications:

As stated in the objectives, the broad fundamental aim of this study is developing computational tools and pipelines for a comprehensive analysis of QS systems, the results of which were presented and discussed in previous sections of this chapter. In addition to the mainly targeted questions, during the course of the study, several spin-off research topics emerged and gave rise to further studies. One of them is comprehensive analysis of solo *luxR*s (described in Chapter 1.6.2) independently from topologies of other AHL based QS system genes. Second one was construction of web based interface for visually presenting AHL QS system topologies for *Burkholderia* genus. And the biggest one is generalization of developed methods to other QS systems, and building a web portal with an aim of automatizing the described annotation tasks to maximum extent and providing researchers with detailed genomic, quantitative and phylogenetic information in interactive web based system. The following chapters will describe above mentioned tasks in given order.

3.5.1. Analysis of solo *luxR* genes

AHL based QS systems are perhaps the best studied and best understood among the bacterial intracellular mechanisms, however the role of the so-called solo *luxR* proteins that are present in many bacteria, are relatively poorly understood. The goal of this project was to employ the computational tools that I helped to develop, to a comprehensive analysis of *luxR* genes in bacterial genomes. One of the novel parts of this analysis is that I also scanned draft genomes for solo *luxR* occurrences.

I wanted to address the following questions: a) Have the solo *luxR*s evolved independently from those *luxR*s that are in well-defined QS circuits (QS topologies). b) Are there specific arrangements of *luxR* solos in the prokaryotic chromosomes? c) Are there novel sequence features in the *luxR* solos?

An important question is to predict whether or not a solo *luxR* protein is likely to bind AHL. This question can be studied only by laboratory experiment, even though a few prediction methods were developed for the purpose. Namely, one can study the 3D structure of *luxR* – AHL complexes and pinpoint the amino acids that are necessary for ligand binding. There are very few such structures available, but there is a small group of amino acids that are a) seen as ligand-binding in the 3D structures and b) are sufficiently conserved in the proteins that are known to be AHL binders. Such residues can be summarized as regular expressions. In addition, the groups of Vittorio Venturi and Doriano Lamba have identified a few residues that are likely to be conserved in non AHL binding solos [94] [95]. Moreover, we have identified a few patterns conserved in *Burkholderia luxR*s. From these we developed a small battery of regular expressions that fall into two groups, AHL binders and non-AHL binders (Table 10).

Census of solo LuxR genes in prokaryotic genomes.

I scanned all prokaryotic genomes present in the NCBI databases as of March 2014. This included 2620 complete and 6970 draft genomes with 644474 annotated and 505155 un-annotated contigs, and a total of over 25 million protein-coding ORFs. I used Hidden Markov Model recognizers that contained an HMM module for autoinducer binding domain and an additional HMM module for the *GerE* DNA binding domain (used data described in Chapter 2.2). This census revealed that there are 64 new occurrences in which the functions were indicated as hypothetical. The complete list of the genera is shown in Table 9.

Genus	Total	Hypothetical
Shigella	44	0
Polaromonas	1	0
Sphingobium	7	2
Erythrobacter	4	0
Bradyrhizobium	7	0
Geobacter	2	0
Caulobacter	4	0
Cupriavidus	3	0
Dickeya	3	0
Chelativorans	1	0
Sagittula	3	1
Pelagibaca	2	0
Methylocella	1	0
Ahrensia	1	0
Xanthobacter	1	0
Oceanicola	4	0
Ochrobactrum	12	0
Burkholderiales	1	0
Pelagibacterium	1	0
Magnetospirillum	1	0
Oceanibulbus	4	0
Micavibrio	1	0
Pseudoalteromonas	1	0
Thalassiobium	1	0
Salmonella	307	3
Polymorphum	2	0
Gluconacetobacter	5	0
Candidatus	2	0
Celeribacter	3	0

Ruegeria	6	0
Acetobacter	1	0
Escherichia	730	3
Desulfurispirillum	1	0
Tolumonas	1	0
Haliangium	1	0
Klebsiella	63	12
Acinetobacter	13	0
Octadecabacter	2	0
Rhodobacteraceae	1	0
Mesorhizobium	15	2
Novosphingobium	3	0
Comamonas	3	0
Sodalis	1	0
Roseibium	2	0
Roseovarius	4	0
Phenylobacterium	1	0
Citreicella	3	0
Collimonas	1	0
Salinibacterium	1	0
Variovorax	3	0
Pseudomonas	254	5
Lutiella	1	0
Acidithiobacillus	1	0
Oxalobacteraceae	1	1
Azospirillum	1	0
Roseobacter	9	0
Leptospirillum	1	0
Sinorhizobium	32	0
Agrobacterium	50	1
Hoeflea	4	0
Sorangium	1	1
Parvibaculum	1	0
Legionella	2	0
Raoultella	1	0
Enterobacteriaceae	1	0
Rhodobacterales	3	0
Cellvibrio	2	0
Brenneria	1	0
Methylibium	1	0
Labrenzia	1	0
Gamma	1	0
Vibrio	116	0
Rahnella	3	0
Ralstonia	4	2

Alcanivorax	1	0
Oceanicaulis	1	0
Aromatoleum	1	1
Sphingomonas	10	0
Hirschia	1	0
Pectobacterium	9	0
Xanthomonas	23	0
Cronobacter	5	0
Methylacidiphilum	2	0
Maritimibacter	1	0
Rubrivivax	2	0
Yersinia	24	0
Phaeobacter	4	0
Beijerinckia	1	0
Nitratireductor	6	0
Burkholderia	179	1
Nitrobacter	3	0
Fulvimarina	1	0
Jannaschia	3	0
Azoarcus	1	1
Oligotropha	5	0
Nitrosospira	1	0
Rhizobium	166	4
Aeromonas	23	8
Serratia	30	0
Pantoea	2	0
Paracoccus	4	0
Phyllobacterium	3	0
Achromobacter	1	0
Frateuria	1	0
Dinoroseobacter	1	0
Caenispirillum	2	0
Citrobacter	8	0
Rhodobacter	13	1
Brucella	109	10
Pseudovibrio	4	2
Rhodospirillum	5	0
Aliivibrio	1	0
Afipia	1	1
Silicibacter	4	0
Sulfitobacter	4	0
Enterobacter	19	0
Rhodopseudomonas	2	0
Stenotrophomonas	8	0
Yokenella	1	0

Aurantimonas	1	0
alpha	1	0
Fluoribacter	1	0
Frankia	1	0
Photorhabdus	2	2
Acidovorax	3	0
Desulfovibrio	1	0
Citromicrobium	2	0
Sideroxydans	1	0
Oceaniovalibus	1	0
Methylobacterium	1	0

Table 9. List of genera for members of which soloRs were found. Many of the recognized luxR genes were annotated as hypothetical.

Among the genera in Table 9 we find a number of occurrences in which *luxR* have not yet been described. Altogether, we have 3514 *luxR* genes (including solos, twin Rs and QS-linked counterparts), 2488 of which are solo *luxR*s, i.e. they are clearly more numerous (2488 occurrences) than their QS-linked counterparts (884 occurrences). The similarity cladograms of all *luxR* sequences is clearly too big to overview. Yet, it is very important to overview it since perhaps the most important question is to decide if the *luxR* solos cluster separately from those in known QS systems. This tendency is perceivable in the large *luxR* tree, but in order to see it more clearly, we restricted the in depth analysis to *Burkholderia* genus first.

The tree of *Burkholderia luxR* proteins is shown in Figure 34. Here we clearly see that a) Also in *Burkholderia*, *luxR* proteins are more numerous ((271 occurrences, out of which 179 are solos) than QS-linked *luxR*s (169 occurrences)). b) Solo *luxR*s form clades (colored blue and red) that are separate from the QS-linked *luxR*s (colored black). The latter form the usual clades denoted by the type of topology (RI, RMI, RXI etc., see section 3.1). From the clades of solo *luxR*s, some of the groups are difficult to interpret. There are a few, however, that deserve special attention.

First we see a group labeled RR which denotes a novel topology that we term twin R arrangement in which two *luxR* proteins are located next to each other. We see two versions of this topology, shown in Figure 35. The divergent topology occurs only in the *Burkholderia* genus. One of the *luxR*s is longer than the other one so we introduced the "short/long" notation for the genes. The two *luxR*s are not immediately vicinal and the intergenic region in some of the annotated genomes contains a short ORF arranged in tandem with the long *luxR* gene. But this ORF is too short so one cannot be certain that it is not an annotation artifact.

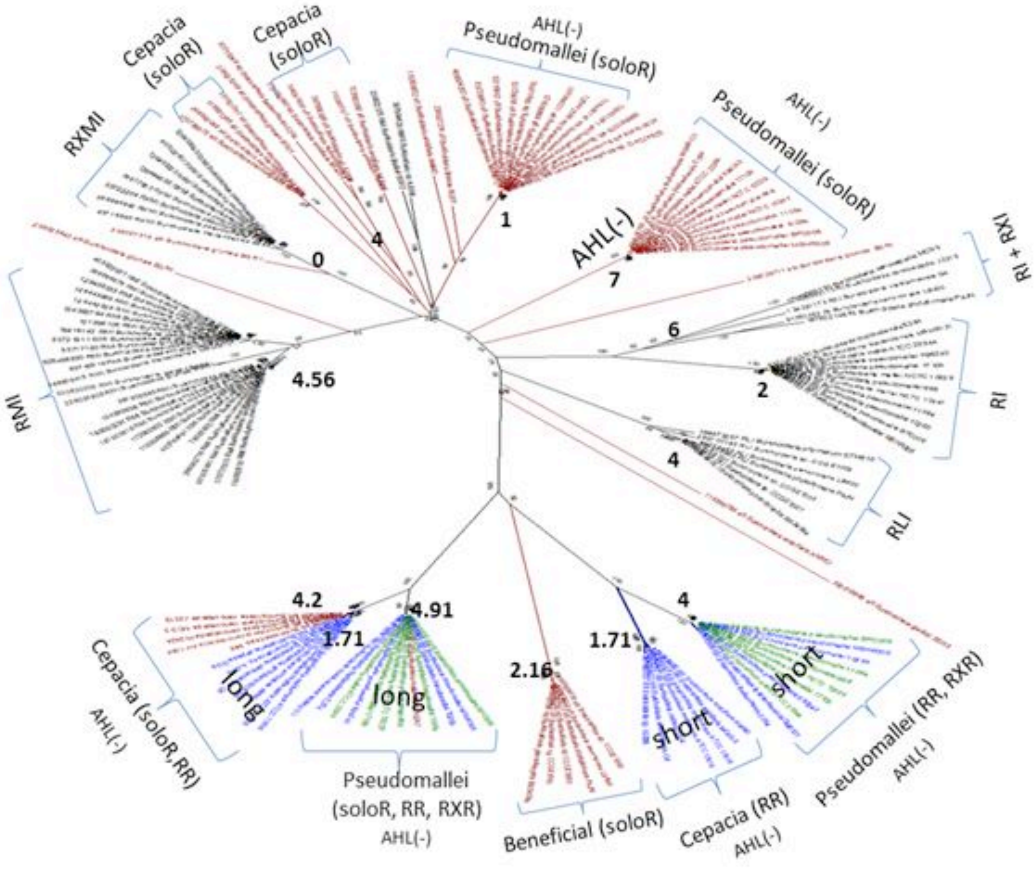
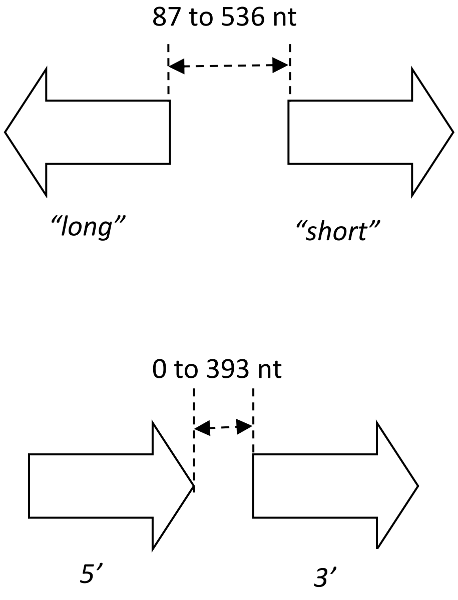


Figure 34. Cladogram of *Burkholderia* according to luxR genes. Black indicates *luxR* in canonical topologies (RI, RMI, RXMI), red indicates solos and blue indicates solos in twin R (RR) topologies. AHL- indicates the lack of AHL binding predicted by regular expression search (Table 11, below)



Divergent topology. Occurs only in *Burkholderia*. The intergenic region sometimes contains a short, hypothetical ORF

Tandem topology. Occurs only outside *Burkholderia* (Caenispirillum Lautropia Methylobacterium Micavibrio Pantoea Rhizobium Sinorhizobium)

Figure 35. Twin *luxR* arrangements and the notations used in this thesis.

Sequence variability

The sequence variability within the *luxR* protein is quite complex. Namely, the sequence of the autoinducer domain is highly variable, sometimes only a few key amino acids are conserved in it. The *GerE* DNA-binding domain on the other hand seems far more conserved in comparison with the autoinducer domain, but we have to remember that the *GerE* domain is a member of a large clan of helix-turn-helix proteins, perhaps the widest class of DNA binders, so its sequence bears similarities to great many other proteins.

For this reason we primarily studied the conservation within the autoinducer domain. The question asked was whether or not one can say sequence features that are characteristic of one or other clade in the *luxR* cladograms. First we identified a number of sequence patterns that are conserved in AHL-binding or non AHL binding autoinducer domains (Table 10)

	AHL-binders
1	Y.(3)W.(3)Y.(8)D.(13,14)W
2	W.(3)Y.(8)DP.(13)W.(32)G
3	W.(3)Y.(8)D[PS].(12,13)W.(32)G
4	W.(3)Y.(8)DP.(13)W.(24-32)G
5	W.(3)Y.(8)D[PS].(12,13)W.(24-32)G
6	Y.{3}W.{3}Y.{8}DP.{13}W.{19}A.{3}G.{3}G
7	Y.{3}W.{3}Y.{8}DP.{13}W.{14}A.{3}G.{3}G
8	[VF].{3}W.{3}Y.{8}DP.{13}W.{14,19}[CR].{3}[GP].{3}G
	Non-AHL binders
11	Y.{3}W.{3}Y.{8}DP.{13}W.{19}A.{3}G.{3}G
10	Y.{3}W.{3}Y.{8}DP.{13}W.{14}A.{3}G.{3}G
11	[VF].{3}W.{3}Y.{8}DP.{13}W.{14,19}[CR].{3}[GP].{3}G

Table 10. Regular expressions. They were found in the autoinducer binding domain of AHL-binding and non-AHL binding *luxR* proteins

Then we used these regular expressions to predict whether or not various groups of *luxR*s are likely to bind AHLs. As shown in the previous table, the regular expressions are partly overlapping, so the results are partly redundant. In order to get a clearer picture, we present the hits grouped according to the solo clades identified in the cladogram of *Burkholderia luxR* proteins.

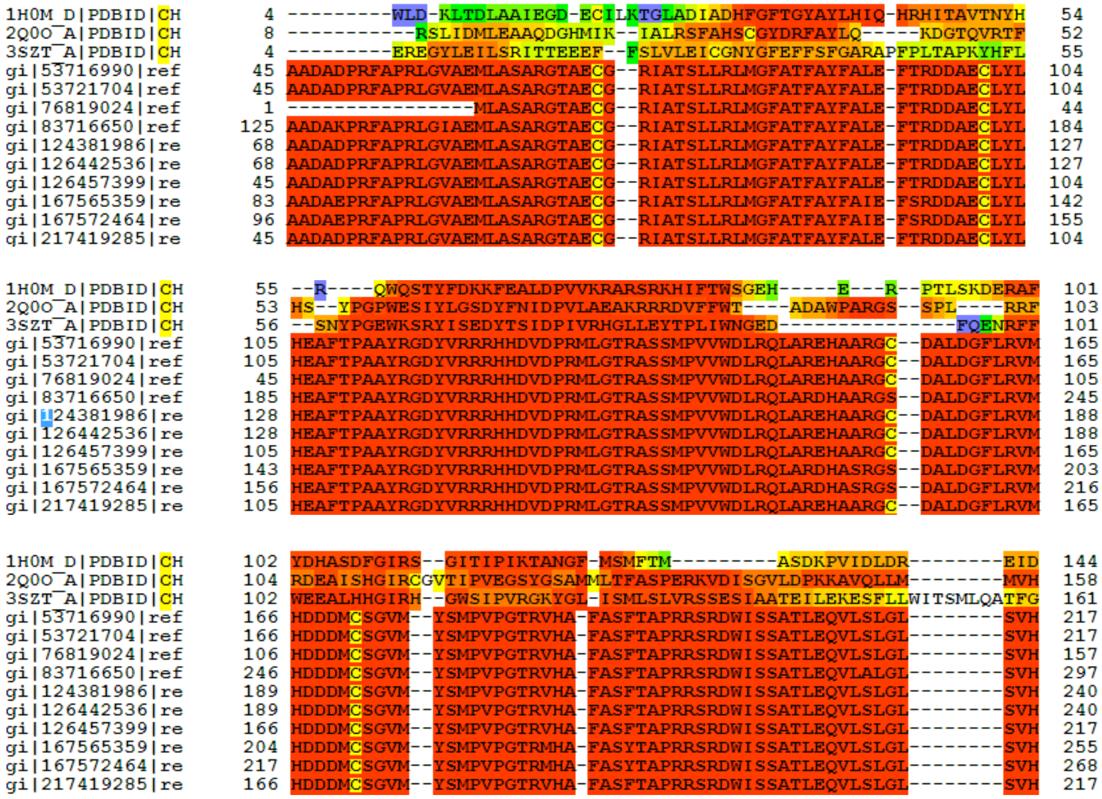
Pattern	AHL								Non-AHL			AHL binding
Clade-name	1	2	3	4	5	6	7	8	9	10	11	
Solo clades:												
cenocepacia 01											+	-
cenocapacia 02		+	+	+	+							+
pseudomallei 01				+	+							+
pseudomallei 02											+	-
pseudomallei 03										+		-
pseudomallei 04	+			+	+							+
pseudomallei 05	+	+	+	+	+		+					+
thailandensis				+	+							+
Twin solo clades												
RRCepacia_long											+	-

RRMallei_long											+	-
RRCepacia_short										+		-
RRMallei_short										+		-
RRnonB_3T	+			+	+							+
RRnonB_5T	+	+	+	+	+							+

Table 11. Prediction of AHL-binding domains in solo *luxR* proteins using regular expressions

The hits found by the regular expressions separated well according to the clades i.e. one clade had hits either with the AHL-binder, or with the non-AHL binder patterns. The corresponding labels were then added to the *luxR* tree.

Another, somewhat unexpected finding was the presence of conserved cysteines in some of the *luxR* clades, an example is shown in Figure 36.



```
1H0M_D|PDBID|CH      4 ---------WLD-KLTDLAAIEGD-ECILKTGLADIADHFGFTGYAYLHIQ-HRHITAVTNYH   54
2Q0O_A|PDBID|CH      8 -----------RSLIDMLEAAQDGHMIK--IALRSFAHSCGYDRFAYLQ-----KDGTQVRTF   52
3SZT_A|PDBID|CH      4 ---------EREGYLEILSRITTEEEF--FSLVLEICGNYGFEFFSFGARAPFPLTAPKYHFL   55
gi|53716990|ref     45 AADADPRFAPRLGVAEMLASARGTAECG--RIATSLLRLMGFATFAYFALE-FTRDDAECLYL  104
gi|53721704|ref     45 AADADPRFAPRLGVAEMLASARGTAECG--RIATSLLRLMGFATFAYFALE-FTRDDAECLYL  104
gi|76819024|ref      1 ----------------MLASARGTAECG--RIATSLLRLMGFATFAYFALE-FTRDDAECLYL   44
gi|83716650|ref    125 AADAKPRFAPRLGIAEMLASARGTAECG--RIATSLLRLMGFATFAYFALE-FTRDDAECLYL  184
gi|124381986|re     68 AADADPRFAPRLGVAEMLASARGTAECG--RIATSLLRLMGFATFAYFALE-FTRDDAECLYL  127
gi|126442536|re     68 AADADPRFAPRLGVAEMLASARGTAECG--RIATSLLRLMGFATFAYFALE-FTRDDAECLYL  127
gi|126457399|re     45 AADADPRFAPRLGVAEMLASARGTAECG--RIATSLLRLMGFATFAYFALE-FTRDDAECLYL  104
gi|167565359|re     83 AADAEPRFAPRLGVAEMLASARGTAECG--RIATSLLRLMGFATFAYFAIE-FSRDDAECLYL  142
gi|167572464|re     96 AADAEPRFAPRLGVAEMLASARGTAECG--RIATSLLRLMGFATFAYFAIE-FSRDDAECLYL  155
gi|217419285|re     45 AADADPRFAPRLGVAEMLASARGTAECG--RIATSLLRLMGFATFAYFALE-FTRDDAECLYL  104

1H0M_D|PDBID|CH     55 --R----QWQSTYFDKKFEALDPVVKRARSRKHIFTWSGEH-----E---R--PTLSKDERAF  101
2Q0O_A|PDBID|CH     53 HS--YPGPWESIYLGSDYFNIDPVLAEAKRRRDVFFWT----ADAWPARGS--SPL----RRF  103
3SZT_A|PDBID|CH     56 --SNYPGEWKSRYISEDYTSIDPIVRHGLLEYTPLIWNGED---------------FQENRFF  101
gi|53716990|ref    105 HEAFTPAAYRGDYVRRRHHDVDPRMLGTRASSMPVVWDLRQLAREHAARGC--DALDGFLRVM  165
gi|53721704|ref    105 HEAFTPAAYRGDYVRRRHHDVDPRMLGTRASSMPVVWDLRQLAREHAARGC--DALDGFLRVM  165
gi|76819024|ref     45 HEAFTPAAYRGDYVRRRHHDVDPRMLGTRASSMPVVWDLRQLAREHAARGC--DALDGFLRVM  105
gi|83716650|ref    185 HEAFTPAAYRGDYVRRRHHDVDPRMLGTRASSMPVVWDLRQLARDHAARGS--DALDGFLRVM  245
gi|124381986|re    128 HEAFTPAAYRGDYVRRRHHDVDPRMLGTRASSMPVVWDLRQLAREHAARGC--DALDGFLRVM  188
gi|126442536|re    128 HEAFTPAAYRGDYVRRRHHDVDPRMLGTRASSMPVVWDLRQLAREHAARGC--DALDGFLRVM  188
gi|126457399|re    105 HEAFTPAAYRGDYVRRRHHDVDPRMLGTRASSMPVVWDLRQLAREHAARGC--DALDGFLRVM  165
gi|167565359|re    143 HEAFTPAAYRGDYVRRRHHDVDPRMLGTRASSMPVVWDLRQLARDHASRGS--DALDGFLRVM  203
gi|167572464|re    156 HEAFTPAAYRGDYVRRRHHDVDPRMLGTRASSMPVVWDLRQLARDHASRGS--DALDGFLRVM  216
gi|217419285|re    105 HEAFTPAAYRGDYVRRRHHDVDPRMLGTRASSMPVVWDLRQLAREHAARGC--DALDGFLRVM  165

1H0M_D|PDBID|CH    102 YDHASDFGIRS--GITIPIKTANGF-MSMFTM---------ASDKPVIDLDR--------EID  144
2Q0O_A|PDBID|CH    104 RDEAISHGIRCGVTIPVEGSYGSAMMLTFASPERKVDISGVLDPKKAVQLLM--------MVH  158
3SZT_A|PDBID|CH    102 WEEALHHGIRH--GWSIPVRGKYGL-ISMLSLVRSSESIAATEILEKESFLLWITSMLQATFG  161
gi|53716990|ref    166 HDDDMCSGVM--YSMPVPGTRVHA-FASFTAPRRSRDWISSATLEQVLSLGL--------SVH  217
gi|53721704|ref    166 HDDDMCSGVM--YSMPVPGTRVHA-FASFTAPRRSRDWISSATLEQVLSLGL--------SVH  217
gi|76819024|ref    106 HDDDMCSGVM--YSMPVPGTRVHA-FASFTAPRRSRDWISSATLEQVLSLGL--------SVH  157
gi|83716650|ref    246 HDDDMCSGVM--YSMPVPGTRVHA-FASFTAPRRSRDWISSATLEQVLALGL--------SVH  297
gi|124381986|re    189 HDDDMCSGVM--YSMPVPGTRVHA-FASFTAPRRSRDWISSATLEQVLSLGL--------SVH  240
gi|126442536|re    189 HDDDMCSGVM--YSMPVPGTRVHA-FASFTAPRRSRDWISSATLEQVLSLGL--------SVH  240
gi|126457399|re    166 HDDDMCSGVM--YSMPVPGTRVHA-FASFTAPRRSRDWISSATLEQVLSLGL--------SVH  217
gi|167565359|re    204 HDDDMCSGVM--YSMPVPGTRMHA-FASYTAPRRSRDWISSATLEQVLSLGL--------SVH  255
gi|167572464|re    217 HDDDMCSGVM--YSMPVPGTRMHA-FASYTAPRRSRDWISSATLEQVLSLGL--------SVH  268
gi|217419285|re    166 HDDDMCSGVM--YSMPVPGTRVHA-FASFTAPRRSRDWISSATLEQVLSLGL--------SVH  217
```

Figure 36. A conserved cysteine patterns (yellow). Depicted are the cases found in the Long. *B. mallei* clade of Twin R proteins. Structural multiple alignment of *luxR*s with three known *luxR* protein 3D structures (top three sequences), using the t-coffee alignment pipeline.

Characteristically, we see cysteine patterns both in QS-linked and in solo *luxR* proteins. Interestingly, cysteine patterns are known to be almost sure indicators of disulfide bridges that are characteristic of secreted proteins. On the other hand, *luxR* is a cytoplasmic protein which is in contact with the chromosome. So we need further evidence to show that these cysteine patterns are just frozen accidents or can in fact be involved in disulfide bridges. Such pieces of circumstantial evidence may be derived from the 3D arrangement of the conserved cysteines. If the conserved cystein positions are within a proper distance, we can predict that the cystein pair in question is involved with intra-domain, intra-molecular, dimerizing or tertramerizing disulfide bonds. These studies are underway at the time of writing of this thesis.

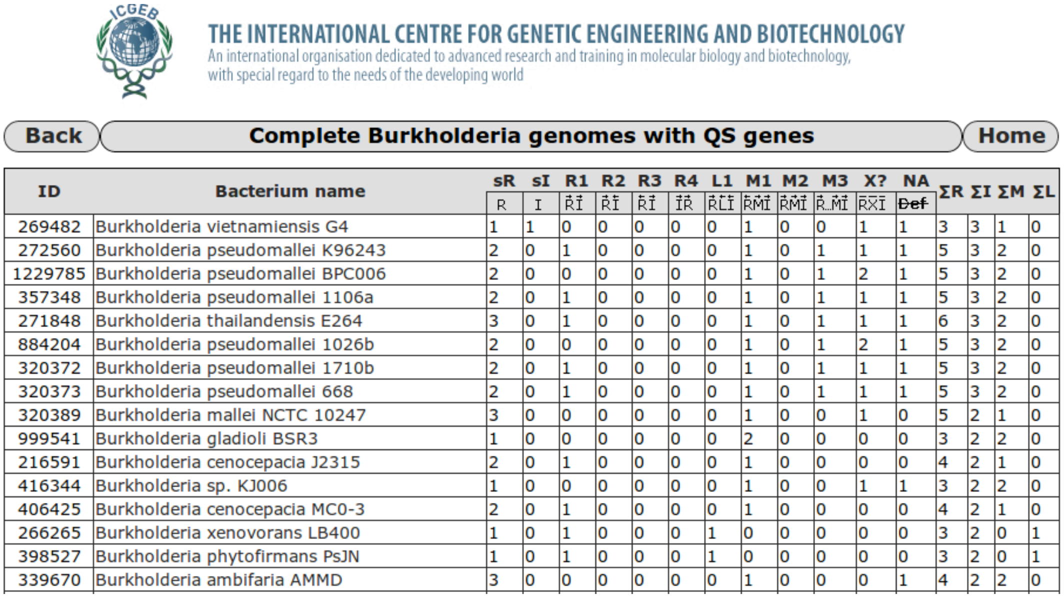
In this chapter I presented results of a comprehensive census of solo *LuxR*-like genes in 2620 complete and 6970 draft prokaryotic genomes (sequenced by the end of 2013). After manually checking the data for false-positive and false-negative hits, we found 2552 *luxR*-like predictions. The census data show that AHL quorum sensing loci solo *luxR* like proteins occur largely in *Proteobacteria*. From a broader perspective, *luxR* proteins belong to a wide class of repressors that contain an autoinducer signal binding domain that binds, covalently or noncovalently a signal molecule, changes its dimerization state and binds to DNA. In the largest class of these molecules, an N-terminal *luxR* autoinducer domain is the signal binder, and a C-terminal *GerE*-type helix-turn helix domain is the DNA binder. However, there are varieties in which the DNA binding domain is a sigma factor, or the N-terminal signaling domain is phosphorylated.

It seems that the number of *luxR* solo genes is higher than the number of QS-linked *luxR* proteins. We built cladogram of *luxR* solo proteins but for clarity, we analyzed the *luxR* proteins of *Burkholderia* in greater detail. We found a novel topological unit that we termed twin *luxR* motif whose transcriptional oritentation can be either divergent, as we see in *Burkholderia*, or tandem, as we see outside the *Burkholderia* genus, notably in *Sinorhizobium, Rhizobium, Methilobacterium.*

Using regular expressions we found that only some of the solo *luxR* proteins are likely to bind AHLs, others may respond to other, hitherto unknown signals.

3.5.2 Presenting AHL QS system data for *Burkholderia*

The findings of this study were part of several publications, and several ongoing other studies. As a part of QS system analysis tasks, a website was developed for presenting detailed information about topologies, chromosomal arrangements and neighborhoods of AHL based QS systems in *Burkholderia*. The website was deployed in ICGEB servers and available from address: http://net.icgeb.org/burkholderia/. It includes the AHL based QS system data mined from complete genomes, draft genomes and individual Genbank entries as well. For each of the data source type (complete/draft genomes and Genbank entries), it provides a table summarizing every *Burkholderia* organism and gives the total number of certain types of topologies found (Figure 37). Each listed genome is summarized further in a separate page, giving the list of topologies found, and link to detailed neighborhoods and chromosomal diagrams pages (Figure 38). Each topology can be browsed in more details. Figure 39 depicts genes which constitute M1 topology (RMI) and the genes from flanking regions. Each line represents a gene, together with annotation data. The first field of the table denotes the ID of chromosome on which the genes are located. Both gene and chromosome IDs are linked to their corresponding NCBI GenBank pages. It is possible to retrieve sequence of gene of interest using the link given in the last column of table.

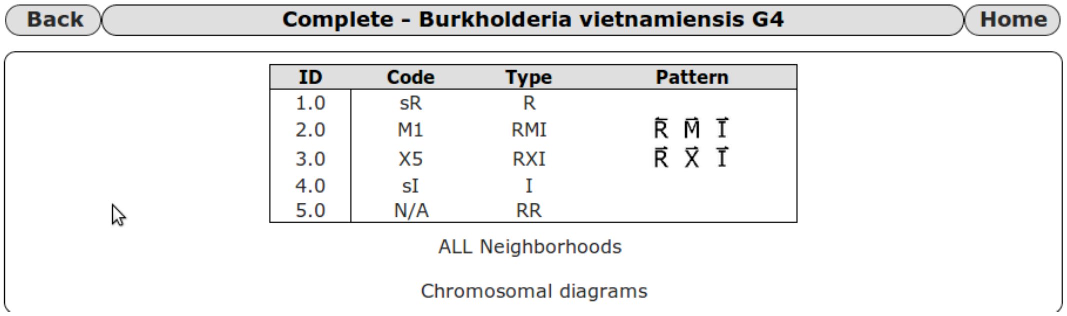


THE INTERNATIONAL CENTRE FOR GENETIC ENGINEERING AND BIOTECHNOLOGY
An international organisation dedicated to advanced research and training in molecular biology and biotechnology,
with special regard to the needs of the developing world

Complete Burkholderia genomes with QS genes

ID	Bacterium name	sR	sI	R1	R2	R3	R4	L1	M1	M2	M3	X?	NA	ΣR	ΣI	ΣM	ΣL
		R	I	$\vec{R}\vec{I}$	$\vec{R}\vec{I}$	$\vec{R}\vec{I}$	$\vec{I}\vec{R}$	$\vec{R}\vec{L}\vec{I}$	$\vec{R}\vec{M}\vec{I}$	$\vec{R}\vec{M}\vec{I}$	$\vec{R}..\vec{M}\vec{I}$	$\bar{R}\bar{X}\bar{I}$	~~Def~~				
269482	Burkholderia vietnamiensis G4	1	1	0	0	0	0	0	1	0	0	1	1	3	3	1	0
272560	Burkholderia pseudomallei K96243	2	0	1	0	0	0	0	1	0	1	1	1	5	3	2	0
1229785	Burkholderia pseudomallei BPC006	2	0	0	0	0	0	0	1	0	1	2	1	5	3	2	0
357348	Burkholderia pseudomallei 1106a	2	0	1	0	0	0	0	1	0	1	1	1	5	3	2	0
271848	Burkholderia thailandensis E264	3	0	1	0	0	0	0	1	0	1	1	1	6	3	2	0
884204	Burkholderia pseudomallei 1026b	2	0	0	0	0	0	0	1	0	1	2	1	5	3	2	0
320372	Burkholderia pseudomallei 1710b	2	0	1	0	0	0	0	1	0	1	1	1	5	3	2	0
320373	Burkholderia pseudomallei 668	2	0	1	0	0	0	0	1	0	1	1	1	5	3	2	0
320389	Burkholderia mallei NCTC 10247	3	0	0	0	0	0	0	1	0	0	1	0	5	2	1	0
999541	Burkholderia gladioli BSR3	1	0	0	0	0	0	0	2	0	0	0	0	3	2	2	0
216591	Burkholderia cenocepacia J2315	2	0	1	0	0	0	0	1	0	0	0	0	4	2	1	0
416344	Burkholderia sp. KJ006	1	0	0	0	0	0	0	1	0	0	1	1	3	2	2	0
406425	Burkholderia cenocepacia MC0-3	2	0	1	0	0	0	0	1	0	0	0	0	4	2	1	0
266265	Burkholderia xenovorans LB400	1	0	1	0	0	0	1	0	0	0	0	0	3	2	0	1
398527	Burkholderia phytofirmans PsJN	1	0	1	0	0	0	1	0	0	0	0	0	3	2	0	1
339670	Burkholderia ambifaria AMMD	3	0	0	0	0	0	0	1	0	0	0	1	4	2	2	0

Figure 37. Complete *Burkholderia* listed with AHL QS genes summarized.



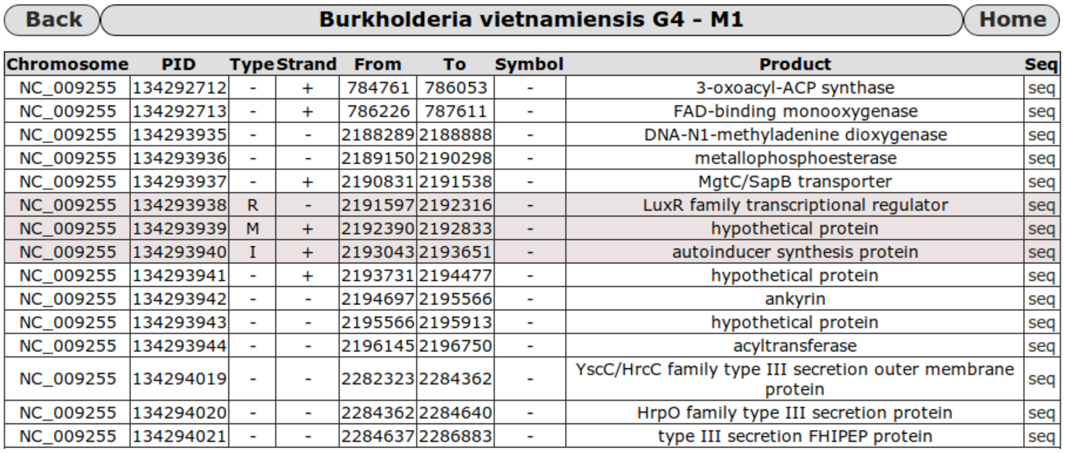
Complete - Burkholderia vietnamiensis G4

ID	Code	Type	Pattern
1.0	sR	R	
2.0	M1	RMI	$\vec{R}\ \vec{M}\ \vec{I}$
3.0	X5	RXI	$\vec{R}\ \vec{X}\ \vec{I}$
4.0	sI	I	
5.0	N/A	RR	

ALL Neighborhoods

Chromosomal diagrams

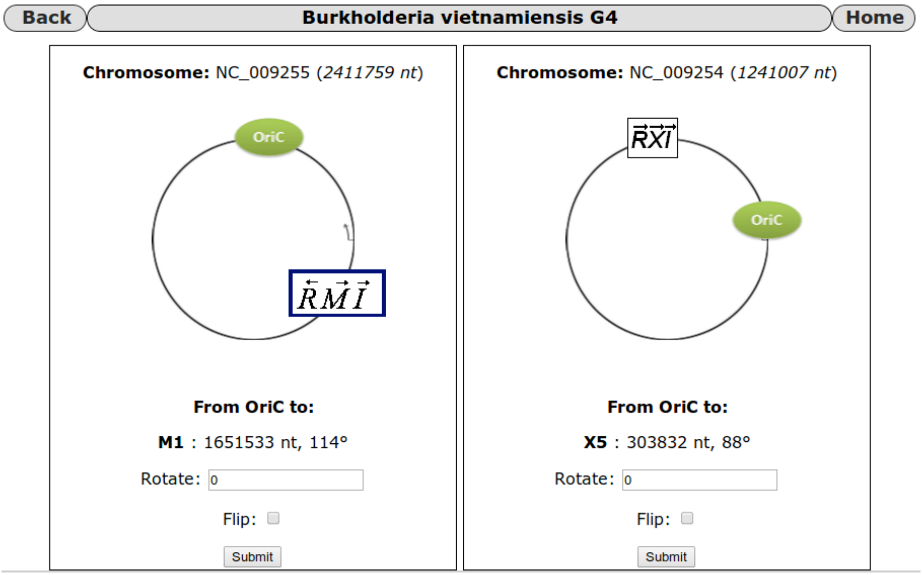
Figure 38. Summary page of *Burkholderia vietnamensis G4*. It has 5 topologies found. Also, links for browsing the neighborhoods and to chromosomal diagrams are given.



Burkholderia vietnamiensis G4 - M1

Chromosome	PID	Type	Strand	From	To	Symbol	Product	Seq
NC_009255	134292712	-	+	784761	786053	-	3-oxoacyl-ACP synthase	seq
NC_009255	134292713	-	+	786226	787611	-	FAD-binding monooxygenase	seq
NC_009255	134293935	-	-	2188289	2188888	-	DNA-N1-methyladenine dioxygenase	seq
NC_009255	134293936	-	-	2189150	2190298	-	metallophosphoesterase	seq
NC_009255	134293937	-	+	2190831	2191538	-	MgtC/SapB transporter	seq
NC_009255	134293938	R	-	2191597	2192316	-	LuxR family transcriptional regulator	seq
NC_009255	134293939	M	+	2192390	2192833	-	hypothetical protein	seq
NC_009255	134293940	I	+	2193043	2193651	-	autoinducer synthesis protein	seq
NC_009255	134293941	-	+	2193731	2194477	-	hypothetical protein	seq
NC_009255	134293942	-	-	2194697	2195566	-	ankyrin	seq
NC_009255	134293943	-	-	2195566	2195913	-	hypothetical protein	seq
NC_009255	134293944	-	-	2196145	2196750	-	acyltransferase	seq
NC_009255	134294019	-	-	2282323	2284362	-	YscC/HrcC family type III secretion outer membrane protein	seq
NC_009255	134294020	-	-	2284362	2284640	-	HrpO family type III secretion protein	seq
NC_009255	134294021	-	-	2284637	2286883	-	type III secretion FHIPEP protein	seq

Figure 39. M1 topology of *Burkholderia vietnamensis G4.* Topology is found on chromosome NC_009255 and its surrounding genes (neighborhood) are listed with detailed annotation information.



Chromosome: NC_009255 (*2411759 nt*)

From OriC to:

M1 : 1651533 nt, 114°

Rotate: 0

Flip: □

Submit

Chromosome: NC_009254 (*1241007 nt*)

From OriC to:

X5 : 303832 nt, 88°

Rotate: 0

Flip: □

Submit

Figure 40. Circular diagrams.

In addition to neighborhood information described above, the website also gives chromosomal arrangements of QS elements on circular diagram. Figure 40 depicts two chromosomes (NC_009255, NC_009254) of *Burkholderia vietnamensis G4*. The chromosomal diagram illustrates all annotated QS elements, origin of replication and starting point of chromosome. To make it more convenient for visual inspection and analy-sis, rotation and flipping functionalities were added. Such functionality is useful for making larger scale survey and spotting common patterns. For instance, location of M1 topology relative to origin of replication is conserved across species in *Burkholderia*. It can be revealed by observing visually as in Figure 41, where it can be seen that locations of M1 topologies found in *Burkholderia cenocepacia*, *Burkholderia mallei*, *Burkholderia multivorans* and *Burkholderia gladioli* relative to origin of replication is conserved.

This website was used as main demonstration of methods and supplementary materials in publication written by Choudhary et.al. [96]

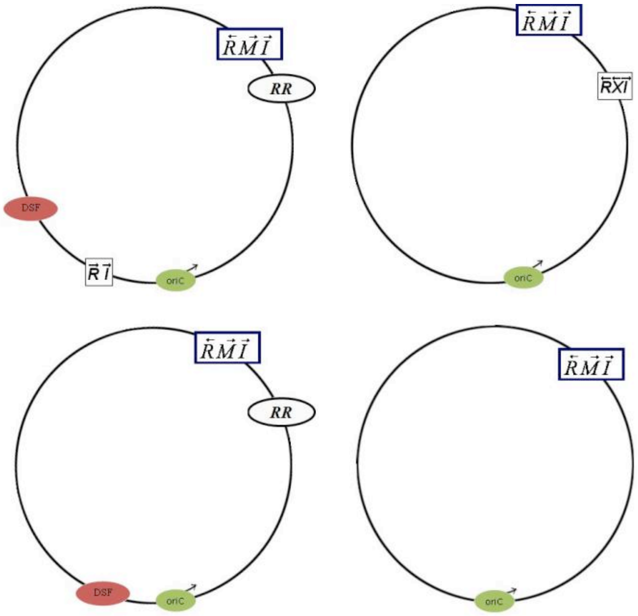


Figure 41. **RMI topology's** in species of *Burkholderia*. Shown chromosomes correspond to genomes of *Burkholderia cenocepacia* (top left), *Burkholderia mallei*(top right), *Burkholderia multivorans*(bottom left) and *Burkholderia gladioli*.

3.5.3 An integrated portal for QS genes

In the last years, there has been an increased amount of researches on QS systems. One of the well-known QS systems is AHL based quorum sensing system, which was the main subject of this study. But, apart from AHL based quorum sensing system, there are number of other known quorum sensing systems, which are relatively less studied. Some of them are:

1) Two-component signal-transduction based quorum sensing systems (in broad sense) [97]

2) Pseudomonas quinolone signaling (PQS) [98]

3) Diffusible signal factor-mediated quorum sensing (DSF) [99]

4) DSF in *Burkholderia* (BDSF) [100]

One idea was to extend the computational tools and methods which were developed during this study to the above mentioned systems. Preliminary searches for genes (Table 13) which constitute these QS systems gave reasonable amounts of hits, which lead to further studies (Table 12).

Name	Hits
Two Component based systems	6333
DSF	6918
BDSF	5131
PQS	452

Table 12. Other types of QS systems, and the number hits we found corresponding to them.

QS system	Regulating genes
Two-Component systems	ABC_Transporters_ComA, ABC_Transporters_blpA, ABC_Transporters_plnG, CQ_AgrB, CQ_and_Pheromone_fsrB_Enterococcus_faecalis, HK_AgrC, HK_blpH, HK_ComD, HK_ComP, HK_PlnB, HK_VirS_Clostridium, Pheromone_AgrD, Pheromone_AgrD_Clostridium_difficile, Pheromone_AgrD_Clostridium_perfringens, Pheromone_AgrD_Listeria_monocytogenes, Pheromone_AgrD_Staphylococcus_saprophyticus, Pheromone_CSPs_ComC,BlpC, Pheromone_lamD_Lactobacillus_plantarum, Pheromone_papR, Pheromone_phrC, Pheromone_PlnA, Pheromone_PltA, Phosphatase_rapC, RR_AgrA, RR_BlpR, RR_BlpS, RR_ComA, RR_ComE, RR_PlcR, RR_PlnC, RR_PlnD, Transmembrane_protein_BlpB, Transmembrane_protein_ComB, Transmembrane_protein_PlnH, X, CQ_ComQ, Pheromone_ComX
DSF	rpfG, rpfB, rpfC, rpfF, rpfH
BDSF	rpfF, rpfR
PQS	pqsB, pqsC, pqsD, pqsE, pqsA

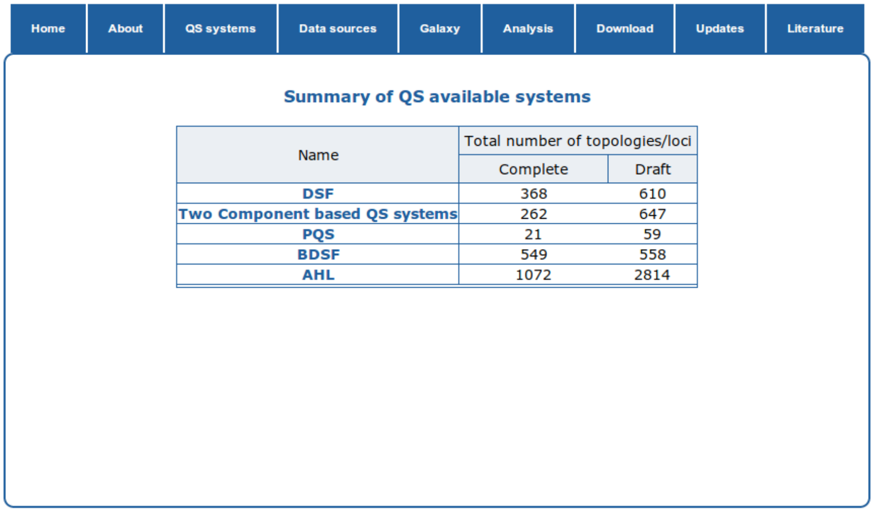
Table 13. Suggested QS systems for extending the framework, and the genes which regulate named systems.

In similar way with *Burkholderia* page described in previous section, this project's aim is to provide detailed information about loci and genes belonging to each of the above mentioned QS systems.

The annotation data can be browsed in different way of categorization. Namely, by:

- QS system's type (Figure 42)
- Data source type (Figure 43)
- Genome wise browsing of assigned QS system types. (Figure 44)

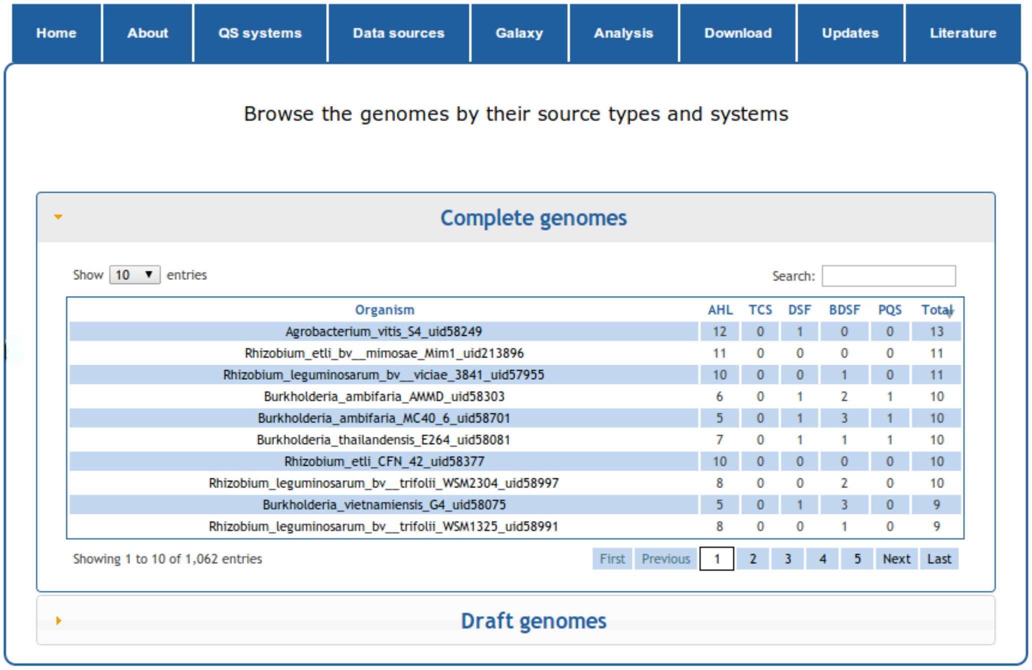
At the time of writing of this thesis, this project is a work in progress. And the modules which are planned to be integrated to it will be described in future works chapter.



Summary of QS available systems

Name	Total number of topologies/loci	
	Complete	Draft
DSF	368	610
Two Component based QS systems	262	647
PQS	21	59
BDSF	549	558
AHL	1072	2814

Figure 42. Summary page of annotation data grouped by QS system type.



Browse the genomes by their source types and systems

Complete genomes

Show 10 ▼ entries

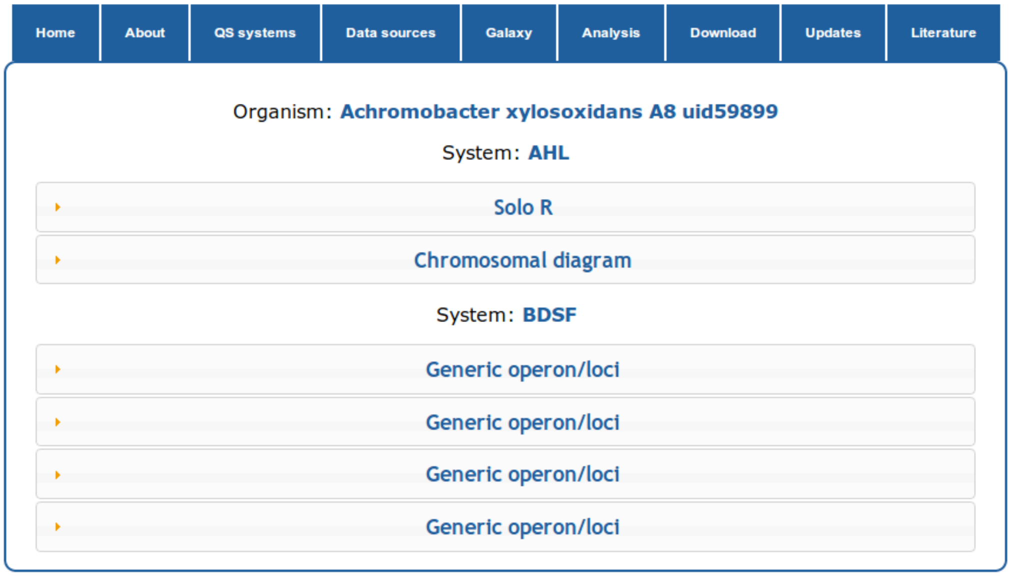
Search:

Organism	AHL	TCS	DSF	BDSF	PQS	Total
Agrobacterium_vitis_S4_uid58249	12	0	1	0	0	13
Rhizobium_etli_bv__mimosae_Mim1_uid213896	11	0	0	0	0	11
Rhizobium_leguminosarum_bv__viciae_3841_uid57955	10	0	0	1	0	11
Burkholderia_ambifaria_AMMD_uid58303	6	0	1	2	1	10
Burkholderia_ambifaria_MC40_6_uid58701	5	0	1	3	1	10
Burkholderia_thailandensis_E264_uid58081	7	0	1	1	1	10
Rhizobium_etli_CFN_42_uid58377	10	0	0	0	0	10
Rhizobium_leguminosarum_bv__trifolii_WSM2304_uid58997	8	0	0	2	0	10
Burkholderia_vietnamiensis_G4_uid58075	5	0	1	3	0	9
Rhizobium_leguminosarum_bv__trifolii_WSM1325_uid58991	8	0	0	1	0	9

Showing 1 to 10 of 1,062 entries

Draft genomes

Figure 43. Genomes listed with the number of QS systems found in them.



Organism: **Achromobacter xylosoxidans A8 uid59899**

System: **AHL**

Solo R

Chromosomal diagram

System: **BDSF**

Generic operon/loci

Generic operon/loci

Generic operon/loci

Generic operon/loci

Figure 44. Organism summary. Each organism is summarized and given information about topologies/loci/genes corresponding to different QS systems, if found.

4. Conclusion, remarks and future work

> "Beware of bugs in the above code; I have only proved it correct, not tried it." – Donald E. Knuth.

4.1 Corollaries from the study

As the cost of sequencing decreases, the amount of publicly available genomic data is growing orders of magnitude faster than it was previously expected. And automated annotation techniques and tools will be cornerstones in processing of this data. This study suggests a subsystem-based annotation approach, and shows that while working on annotation aimed tools, the same tools can be used to address different biological questions.

This study provides complete census of AHL based QS system's *luxI* and *luxR* based topologies, together with their phylogenetic interpretations.

Previously, it was shown that *luxR* genes were unlikely to be acquired by horizontal gene transfers in *Vibrionaceae* [93]. With the comprehensive analysis of *luxR* genes in search for horizontal gene transfers, we generalize the test and confirm the findings of the study for other genera of Bacteria where *luxR* genes were found.

4.2 Tools, environments and languages used

Throughout this study, several programming languages and environments were used. In addition to standard bioinformatics tools described in the introduction chapter, Python was used for scripting. Python was used not only as a tool for scripting, it was also used extensively in calculations and pipelining. For this purposes, it was seen that BioPython and Numpy libraries are mature enough for making advanced calculations.

The main calculations for finding and annotating topologies of QS systems were carried out using Galaxy framework. Tools developed for that task were wrapped as *Galaxy tools*.

Calculations for horizontal gene transfers were done using either MATLAB or Python (Numpy framework). In terms of running time cost, the most expensive task was the calculation of word frequencies in sliding window algorithm. For optimizing this task, the straightforward way of calculating the frequencies was altered. Initially algorithm slices each window from the long DNA sequence and calculates the *k-mer* frequencies over sliced window, which means that certain sequence segments would be counted multiple times for multiple windows, depending on window and offset lengths. Instead, the new approach scans over sequence word-by-word and each time increases the corresponding pointer in all of the windows which span the word being focused. As a result, the algorithm is of *O(N)* complexity with respect to the length of the sequence (Figure 45). The draw-back of this scheme is that, it will require annotation of dictionary (a data structure used for keeping frequency numbers) for each possible window at the beginning of procedure, which will be kept in RAM memory until the overall counting process is over. This might be issue if multiple processes are being run using shared RAM memory.

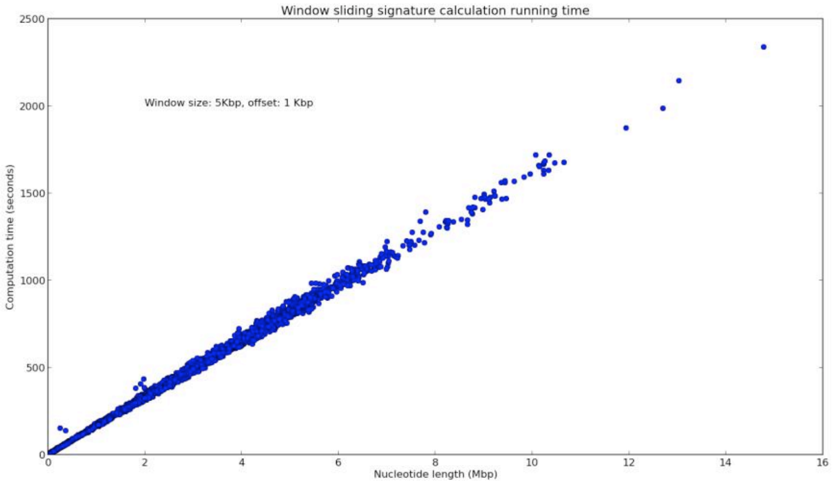


Figure 45. Running time window sliding algorithm vs. length of sequences.

The *Burkholderia* web page described in Chapter 3.6.2 and ongoing QS portal described in Chapter 3.6.3 were both built using Python based Django framework and MySQL relational database management system on the backend, and HTML5, CSS3, JavaScript(jQuery) on the front end.

Relational database scheme which backed the system of *Burkholderia* page was extended for QS portal with new logical units in order to handle larger scale representation (Figure 46).

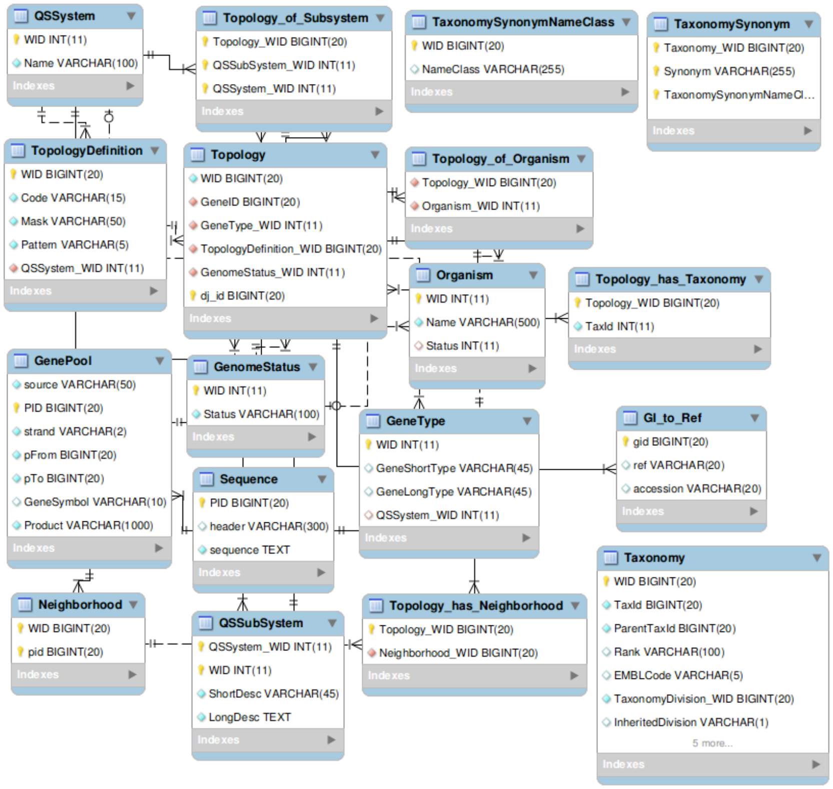


Figure 46. ER diagram of tables which run on the backend of QS portal project.

The database scheme deployed for running QS portal, was used not only for the purpose of running the website, but also for extracting the statistics and detailed information during the study for other projects as well. The flexibility of relational database system allows one to be able to ask interesting questions to the database. For instance, for retrieving solo *luxR* genes described in Chapter 3.6.1, the following SQL query was used:

```sql
select gp.source, gr.Ref, gp.pFrom, gp.pTo, gp.strand
from Topology t
inner join TopologyDefinition td on t.TopologyDefinition_WID = td.WID
inner join GenePool gp on t.GeneID = gp.PID
inner join GI_to_Ref gr on gr.gid = t.GeneID
inner join Topology_of_Organism too on too.Topology_WID = t.WID
inner join Organism o on too.Organism_WID = o.WID
where td.Code='sR'
```

Relatively more complex question “Give number of occurrences for each AHL QS system topology type for genus *Burkholderia, Pseudomonas* have and in total” (Table 5) can be answered with the following SQL command:

```sql
select td.Code,
       coalesce(burk.cnt,0) as 'Burkholderia',
       coalesce(pseud.cnt,0) as 'Pseudomonas',
       coalesce(all_gnm.cnt,0) as 'All'
from TopologyDefinition td
left join (
            select td.WID, count(*) cnt
            from Topology t
            inner join TopologyDefinition td on td.WID = t.TopologyDefinition_WID
            inner join Topology_of_Organism too on too.Topology_WID=t.WID
            inner join Organism o on o.WID = too.Organism_WID
            where o.Name like '%Burkholderia%' and td.QSSystem_WID=1
            group by td.WID
            ) burk on burk.WID = td.WID
left join (
            select td.WID, count(*) cnt
            from Topology t
            inner join TopologyDefinition td on td.WID = t.TopologyDefinition_WID
            inner join Topology_of_Organism too on too.Topology_WID=t.WID
            inner join Organism o on o.WID = too.Organism_WID
            where o.Name like '%Pseudomonas%' and td.QSSystem_WID=1
            group by td.WID
            ) pseud on pseud.WID = td.WID
left join (
            select td.WID, count(*) cnt
            from Topology t
            inner join TopologyDefinition td on td.WID = t.TopologyDefinition_WID
            inner join Topology_of_Organism too on too.Topology_WID=t.WID
            inner join Organism o on o.WID = too.Organism_WID
            where td.QSSystem_WID=1
            group by td.WID
            ) all_gnm on all_gnm.WID=td.WID
where td.QSSystem_WID=1 and td.WID<>5
```

4.3 Future work

4.3.1. Horizontal gene transfer

One of the projects still running at the time of writing of this thesis is development of more comprehensive computational framework for analysis of horizontal gene transfers. Particularly, I plan to try and evaluate different mathematical methods for horizontal transfer inference [67][101][102] which are studied in broader sense, and integrate them to subsystem based annotation approach, studied in this research. It has been shown that bacterial metabolic pathways evolved adaptively mainly due to horizontal gene transfers [103], and the possibility of horizontal gene transfer is one of the first questions that microbiologists ask. Therefore analyzing the growing body of annotated QS systems for horizontal transfer remains an important task.

4.3.2. Quorum Sensing portal

I consider ongoing project of building large scale portal for different Quorum Sensing systems as the main inertia of this study. Some of anticipated additional features will be:

- Comprehensive phylogenetic analysis of several QS systems
- Direct integration of portal with locally deployed Galaxy Framework.
- Automatization of QS system analysis whenever a new genome is sequenced
- Integration of genomics with metagenomic servers.
- Conducting selection analysis on homolog genes of QS system gene types.

While main purposes listed above are kept strict with definition of biologically justified research projects, there are technical cases which can be of great use for QS researchers lacking computational technical skills. For instance, for being able to answer the example questions demonstrated in Chapter 4.2, one would need to have SQL query writing skills and more importantly, familiarity with the relational database scheme used in this study and the tables. To make these kinds of information more accessible to researchers, one additional idea is to create a simpler query builder (within Galaxy framework), which will make it possible to query meaningful questions without knowing underlying technical details. Currently, the gene portal contains many data that await manual curation and assisting this procedure with computational tools is an important task for the future.

5. References

[1] R. D. Fleischmann, M. D. Adams, O. White, R. A. Clayton, E. F. Kirkness, A. R. Kerlavage, C. J. Bult, J. F. Tomb, B. A. Dougherty, J. M. Merrick, and et al., "Whole-genome random sequencing and assembly of Haemophilus influenzae Rd," *Science (80-.).*, vol. 269, no. 5223, pp. 496–512, 1995.

[2] "Genome online database (gold) website http://genomesonline.org."

[3] F. Sanger and A. R. Coulson, "A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase," *J Mol Biol*, vol. 94, no. 3, pp. 441–448, 1975.

[4] R. Cullum, O. Alder, and P. A. Hoodless, "The next generation: using new sequencing technologies to analyse gene regulation," *Respirology*, vol. 16, no. 2, pp. 210–222, 2011.

[5] E. R. Mardis, "A decade's perspective on DNA sequencing technology," *Nature*, vol. 470, no. 7333, pp. 198–203, 2011.

[6] D. A. Benson, M. Cavanaugh, K. Clark, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and E. W. Sayers, "GenBank.," *Nucleic Acids Res.*, vol. 41, no. Database issue, pp. D36–42, Jan. 2013.

[7] "http://www.genome.gov/sequencingcosts."

[8] "NCBI-GenBank Flat File Release 196.0," 2013.

[9] P. M. Nadkarni, L. Marenco, R. Chen, E. Skoufos, G. Shepherd, and P. Miller, "Organization of heterogeneous scientific data using the EAV/CR representation.," *J. Am. Med. Inform. Assoc.*, vol. 6, no. 6, pp. 478–93.

[10] R. Apweiler, A. Bairoch, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, M. J. Martin, D. A. Natale, C. O'Donovan, N. Redaschi, and L.-S. L. Yeh, "UniProt: the Universal Protein knowledgebase.," *Nucleic Acids Res.*, vol. 32, no. Database issue, pp. D115–9, Jan. 2004.

[11] B. E. Suzek, H. Huang, P. McGarvey, R. Mazumder, and C. H. Wu, "UniRef: comprehensive and non-redundant UniProt reference clusters.," *Bioinformatics*, vol. 23, no. 10, pp. 1282–8, May 2007.

[12] B. Boeckmann, A. Bairoch, R. Apweiler, M.-C. Blatter, A. Estreicher, E. Gasteiger, M. J. Martin, K. Michoud, C. O'Donovan, I. Phan, S. Pilbout, and M. Schneider, "The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003.," *Nucleic Acids Res.*, vol. 31, no. 1, pp. 365–70, Jan. 2003.

[13] R. Overbeek, T. Begley, R. M. Butler, J. V Choudhuri, H. Y. Chuang, M. Cohoon, V. de Crecy-Lagard, N. Diaz, T. Disz, R. Edwards, M. Fonstein, E. D. Frank, S. Gerdes, E. M. Glass, A. Goesmann, A. Hanson, D. Iwata-Reuyl, R. Jensen, N. Jamshidi, L. Krause, M. Kubal, N. Larsen, B. Linke, A. C. McHardy, F. Meyer, H. Neuweger, G. Olsen, R. Olson, A. Osterman, V. Portnoy, G. D. Pusch, D. A. Rodionov, C. Ruckert, J. Steiner, R. Stevens, I. Thiele, O. Vassieva, Y. Ye, O. Zagnitko, and V. Vonstein, "The subsystems approach to genome annotation and its use in the project to annotate 1000 genomes," *Nucleic Acids Res*, vol. 33, no. 17, pp. 5691–5702, 2005.

[14] S. Henikoff and J. G. Henikoff, "Amino acid substitution matrices from protein blocks.," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 89, no. 22, pp. 10915–9, Nov. 1992.

[15] R. M. S. M. O. Dayhoff, "Chapter 22: A model of evolutionary change in proteins."

[16] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *J. Mol. Biol.*, vol. 48, no. 3, pp. 443–453, Mar. 1970.

[17] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences.," *J. Mol. Biol.*, vol. 147, no. 1, pp. 195–7, Mar. 1981.

[18] L. WANG and T. JIANG, "On the Complexity of Multiple Sequence Alignment," *J. Comput. Biol.*, vol. 1, no. 4, pp. 337–348, Jan. 1994.

[19] N. Saitou and M. Nei, "The neighbor-joining method: a new method for reconstructing phylogenetic trees," *Mol. Biol. Evol.*, vol. 4, no. 4, pp. 406–425, Jul. 1987.

[20] "A statistical method for evaluating systematic relationships : Free Download & Streaming : Internet Archive." [Online]. Available: http://archive.org/details/cbarchive_33927_astatisticalmethodforevaluatin1902. [Accessed: 12-Jul-2013].

[21] D. G. Higgins and P. M. Sharp, "CLUSTAL: a package for performing multiple sequence alignment on a microcomputer," *Gene*, vol. 73, no. 1, pp. 237–244, Dec. 1988.

[22] J. D. Thompson, D. G. Higgins, and T. J. Gibson, "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice.," *Nucleic Acids Res.*, vol. 22, no. 22, pp. 4673–80, Nov. 1994.

[23] J. D. Thompson, T. J. Gibson, and D. G. Higgins, "Multiple sequence alignment using ClustalW and ClustalX," *Curr Protoc Bioinforma.*, vol. Chapter 2, p. Unit 2 3, 2002.

[24] A. K. Durbin, R. , Eddy, S.R., *Biological Sequence Analysis Probabilistic Models Proteins And Nucleic Acids*. 1998.

[25] S. R. Eddy, "Hidden Markov models," *Curr Opin Struct Biol*, vol. 6, no. 3, pp. 361–365, 1996.

[26] S. R. Eddy, "What is a hidden Markov model?," *Nat Biotechnol*, vol. 22, no. 10, pp. 1315–1316, 2004.

[27] S. R. Eddy, "A new generation of homology search tools based on probabilistic inference," *Genome Inf.*, vol. 23, no. 1, pp. 205–211, 2009.

[28] M. Punta, P. C. Coggill, R. Y. Eberhardt, J. Mistry, J. Tate, C. Boursnell, N. Pang, K. Forslund, G. Ceric, J. Clements, A. Heger, L. Holm, E. L. L. Sonnhammer, S. R. Eddy, A. Bateman, and R. D. Finn, "The Pfam protein families database.," *Nucleic Acids Res.*, vol. 40, no. Database issue, pp. D290–301, Jan. 2012.

[29] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *J Mol Biol*, vol. 215, no. 3, pp. 403–410, 1990.

[30] S. L. Salzberg, A. L. Delcher, S. Kasif, and O. White, "Microbial gene identification using interpolated Markov models.," *Nucleic Acids Res.*, vol. 26, no. 2, pp. 544–8, Jan. 1998.

[31] T. Davidsen, E. Beck, A. Ganapathy, R. Montgomery, N. Zafar, Q. Yang, R. Madupu, P. Goetz, K. Galinsky, O. White, and G. Sutton, "The comprehensive microbial resource.," *Nucleic Acids Res.*, vol. 38, no. Database issue, pp. D340–5, Jan. 2010.

[32] R. L. Tatusov, E. V Koonin, and D. J. Lipman, "A genomic perspective on protein families," *Science (80-.).*, vol. 278, no. 5338, pp. 631–637, 1997.

[33] A. Kuzniar, R. C. H. J. van Ham, S. Pongor, and J. A. M. Leunissen, "The quest for orthologs: finding the corresponding gene across genomes.," *Trends Genet.*, vol. 24, no. 11, pp. 539–51, Nov. 2008.

[34] A. Bairoch, "PROSITE: a dictionary of sites and patterns in proteins," *Nucleic Acids Res*, vol. 19 Suppl, pp. 2241–2245, 1991.

[35] M. Gribskov, "Profile Analysis: Detection of Distantly Related Proteins," *Proc. Natl. Acad. Sci.*, vol. 84, no. 13, pp. 4355–4358, Jul. 1987.

[36] L. Falquet, M. Pagni, P. Bucher, N. Hulo, C. J. Sigrist, K. Hofmann, and A. Bairoch, "The PROSITE database, its status in 2002," *Nucleic Acids Res*, vol. 30, no. 1, pp. 235–238, 2002.

[37] S. Dhir, M. Pacurar, D. Franklin, Z. Gaspari, A. Kertesz-Farkas, A. Kocsor, F. Eisenhaber, and S. Pongor, "Detecting atypical examples of known domain types by sequence similarity searching: the SBASE domain library approach," *Curr Protein Pept Sci*, vol. 11, no. 7, pp. 538–549, 2010.

[38] R. L. Tatusov, N. D. Fedorova, J. D. Jackson, A. R. Jacobs, B. Kiryutin, E. V Koonin, D. M. Krylov, R. Mazumder, S. L. Mekhedov, A. N. Nikolskaya, B. S. Rao, S. Smirnov, A. V Sverdlov, S. Vasudevan, Y. I. Wolf, J. J. Yin, and D. A. Natale, "The COG database: an updated version includes eukaryotes," *BMC Bioinformatics*, vol. 4, p. 41, 2003.

[39] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock, "Gene ontology: tool for the unification of biology. The Gene Ontology Consortium," *Nat Genet*, vol. 25, no. 1, pp. 25–29, 2000.

[40] "http://www.ncbi.nlm.nih.gov/COG/."

[41] L. J. Jensen, P. Julien, M. Kuhn, C. von Mering, J. Muller, T. Doerks, and P. Bork, "eggNOG: automated construction and annotation of orthologous groups of genes.," *Nucleic Acids Res.*, vol. 36, no. Database issue, pp. D250–4, Jan. 2008.

[42] B. L. Bassler, "Small talk. Cell-to-cell communication in bacteria.," *Cell*, vol. 109, no. 4, pp. 421–4, May 2002.

[43] M. J. Stone and D. H. Williams, "On the evolution of functional secondary metabolites (natural products).," *Mol. Microbiol.*, vol. 6, no. 1, pp. 29–34, Jan. 1992.

[44] L. C. Vining, "Roles of secondary metabolites from microbes.," *Ciba Found. Symp.*, vol. 171, pp. 184–94; discussion 195–8, Jan. 1992.

[45] A. L. Demain, "Microbial secondary metabolism: a new theoretical frontier for academia, a new opportunity for industry.," *Ciba Found. Symp.*, vol. 171, pp. 3–16; discussion 16–23, Jan. 1992.

[46] C. Fuqua, M. R. Parsek, and E. P. Greenberg, "Regulation of gene expression by cell-to-cell communication: acyl-homoserine lactone quorum sensing," *Annu Rev Genet*, vol. 35, pp. 439–468, 2001.

[47] W. C. Fuqua, S. C. Winans, and E. P. Greenberg, "Quorum sensing in bacteria: the LuxR-LuxI family of cell density-responsive transcriptional regulators," *J Bacteriol*, vol. 176, no. 2, pp. 269–275, 1994.

[48] J. R. Chandler, S. Heilmann, J. E. Mittler, and E. P. Greenberg, "Acyl-homoserine lactone-dependent eavesdropping promotes competition in a laboratory co-culture model.," *ISME J.*, vol. 6, no. 12, pp. 2219–28, Dec. 2012.

[49] G. H. Wadhams and J. P. Armitage, "Making sense of it all: bacterial chemotaxis.," *Nat. Rev. Mol. Cell Biol.*, vol. 5, no. 12, pp. 1024–37, Dec. 2004.

[50] M. Y. Galperin, "A census of membrane-bound and intracellular signal transduction proteins in bacteria: bacterial IQ, extroverts and introverts.," *BMC Microbiol.*, vol. 5, p. 35, Jan. 2005.

[51] M. Ansaldi and D. Dubnau, "Diversifying selection at the Bacillus quorum-sensing locus and determinants of modification specificity during synthesis of the ComX pheromone.," *J. Bacteriol.*, vol. 186, no. 1, pp. 15–21, Jan. 2004.

[52] A. B. Goryachev, "Design principles of the bacterial quorum sensing gene networks," *Wiley Interdiscip Rev Syst Biol Med*, vol. 1, no. 1, pp. 45–60, 2009.

[53] A. B. Goryachev, "Understanding bacterial cell-cell communication with computational modeling.," *Chem. Rev.*, vol. 111, no. 1, pp. 238–50, Jan. 2011.

[54] N. Keller and T. Hohn, "Metabolic Pathway Gene Clusters in Filamentous Fungi," *Fungal Genet. Biol.*, vol. 21, no. 1, pp. 17–29, Feb. 1997.

[55] A. V Patankar and J. E. González, "Orphan LuxR regulators of quorum sensing.," *FEMS Microbiol. Rev.*, vol. 33, no. 4, pp. 739–56, Jul. 2009.

[56] S. Subramoni and V. Venturi, “LuxR-family ‘solos’: bachelor sensors/regulators of signalling molecules,” *Microbiology*, vol. 155, no. Pt 5, pp. 1377–1385, 2009.

[57] M. Whiteley, K. M. Lee, and E. P. Greenberg, “Identification of genes controlled by quorum sensing in Pseudomonas aeruginosa.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 96, no. 24, pp. 13904–9, Nov. 1999.

[58] C. W. Knapp, “M. Pilar Francino (ed): Horizontal gene transfer in microorganisms,” *Ecotoxicology*, vol. 22, no. 9, pp. 1443–1444, Aug. 2013.

[59] T. Akiba, K. Koyama, Y. Ishiki, S. Kimura, and T. Fukushima, “On the mechanism of the development of multiple-drug-resistant clones of Shigella,” *Trends Microbiol.*, vol. 4, no. 2, pp. 219–227, Feb. 1960.

[60] E. V Koonin, K. S. Makarova, and L. Aravind, “Horizontal gene transfer in prokaryotes: quantification and classification.,” *Annu. Rev. Microbiol.*, vol. 55, pp. 709–42, Jan. 2001.

[61] E. V Koonin and Y. I. Wolf, “Evolution of microbes and viruses: a paradigm shift in evolutionary biology?,” *Front. Cell. Infect. Microbiol.*, vol. 2, p. 119, Jan. 2012.

[62] L. Liu, X. Chen, G. Skogerbø, P. Zhang, R. Chen, S. He, and D.-W. Huang, “The human microbiome: a hot spot of microbial horizontal gene transfer.,” *Genomics*, vol. 100, no. 5, pp. 265–70, Nov. 2012.

[63] L. D. McDaniel, E. Young, J. Delaney, F. Ruhnau, K. B. Ritchie, and J. H. Paul, “High frequency of horizontal gene transfer in the oceans.,” *Science*, vol. 330, no. 6000, p. 50, Oct. 2010.

[64] T. W. Schoener, “The newest synthesis: understanding the interplay of evolutionary and ecological dynamics.,” *Science*, vol. 331, no. 6016, pp. 426–9, Jan. 2011.

[65] J. N. Thompson, *The Geographic Mosaic of Coevolution*. 2005.

[66] J. N. Thompson, “The coevolving web of life.,” *Am. Nat.*, vol. 173, no. 2, pp. 125–40, Feb. 2009.

[67] H. Ochman, J. G. Lawrence, and E. A. Groisman, “Lateral gene transfer and the nature of bacterial innovation.,” *Nature*, vol. 405, no. 6784, pp. 299–304, May 2000.

[68] R. K. Azad and J. G. Lawrence, “Detecting laterally transferred genes.,” *Methods Mol. Biol.*, vol. 855, pp. 281–308, Jan. 2012.

[69] S. Pongor, *Recombinant DNA Part E*, vol. 154. Elsevier, 1987, pp. 450–473.

[70] S. Karlin and I. Ladunga, “Comparisons of eukaryotic genomic sequences.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 91, no. 26, pp. 12832–6, Dec. 1994.

[71] S. Karlin, I. Ladunga, and B. E. Blaisdell, “Heterogeneity of genomes: measures and values.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 91, no. 26, pp. 12837–41, Dec. 1994.

[72] I. Brukner, R. Sánchez, D. Suck, and S. Pongor, “Trinucleotide models for DNA bending propensity: comparison of models based on DNaseI digestion and nucleosome packaging data.,” *J. Biomol. Struct. Dyn.*, vol. 13, no. 2, pp. 309–17, Oct. 1995.

[73] K. Vlahovicek, L. Kaján, and S. Pongor, “DNA analysis servers: plot.it, bend.it, model.it and IS.,” *Nucleic Acids Res.*, vol. 31, no. 13, pp. 3686–7, Jul. 2003.

[74] C. Dufraigne, B. Fertil, S. Lespinats, A. Giron, and P. Deschavanne, “Detection and characterization of horizontal transfers in prokaryotes using genomic signature.,” *Nucleic Acids Res.*, vol. 33, no. 1, p. e6, Jan. 2005.

[75] S. Ménigaud, L. Mallet, G. Picord, C. Churlaud, A. Borrel, and P. Deschavanne, “GOHTAM: a website for ‘Genomic Origin of Horizontal Transfers, Alignment and Metagenomics’.,” *Bioinformatics*, vol. 28, no. 9, pp. 1270–1, May 2012.

[76] C. Chapus, C. Dufraigne, S. Edwards, A. Giron, B. Fertil, and P. Deschavanne, “Exploration of phylogenetic data using a global sequence analysis method.,” *BMC Evol. Biol.*, vol. 5, no. 1, p. 63, Jan. 2005.

[77] P. J. Deschavanne, A. Giron, J. Vilain, G. Fagot, and B. Fertil, “Genomic signature: characterization and classification of species assessed by chaos game representation of sequences.,” *Mol. Biol. Evol.*, vol. 16, no. 10, pp. 1391–9, Oct. 1999.

[78] A. L. Delcher, S. Kasif, R. D. Fleischmann, J. Peterson, O. White, and S. L. Salzberg, “Alignment of whole genomes.,” *Nucleic Acids Res.*, vol. 27, no. 11, pp. 2369–76, Jun. 1999.

[79] A. Boc, A. B. Diallo, and V. Makarenkov, “T-REX: a web server for inferring, validating and visualizing phylogenetic trees and networks.,” *Nucleic Acids Res.*, vol. 40, no. Web Server issue, pp. W573–9, Jul. 2012.

[80] S. Garcia-Vallve, E. Guzman, M. A. Montero, and A. Romeu, “HGT-DB: a database of putative horizontally transferred genes in prokaryotic complete genomes.,” *Nucleic Acids Res.*, vol. 31, no. 1, pp. 187–9, Jan. 2003.

[81] G. S. Vernikos and J. Parkhill, “Interpolated variable order motifs for identification of horizontally acquired DNA: revisiting the Salmonella pathogenicity islands.,” *Bioinformatics*, vol. 22, no. 18, pp. 2196–203, Sep. 2006.

[82] K. A. Frazer, L. Pachter, A. Poliakov, E. M. Rubin, and I. Dubchak, “VISTA: computational tools for comparative genomics.,” *Nucleic Acids Res.*, vol. 32, no. Web Server issue, pp. W273–9, Jul. 2004.

[83] T. J. Carver, K. M. Rutherford, M. Berriman, M.-A. Rajandream, B. G. Barrell, and J. Parkhill, “ACT: the Artemis Comparison Tool.,” *Bioinformatics*, vol. 21, no. 16, pp. 3422–3, Aug. 2005.

[84] J. Oberto, “SyntTax: a web server linking synteny to prokaryotic taxonomy.,” *BMC Bioinformatics*, vol. 14, p. 4, Jan. 2013.

[85] S. M. Kiełbasa, R. Wan, K. Sato, P. Horton, and M. C. Frith, “Adaptive seeds tame genomic sequence comparison.,” *Genome Res.*, vol. 21, no. 3, pp. 487–93, Mar. 2011.

[86] E. Lyons, B. Pedersen, J. Kane, M. Alam, R. Ming, H. Tang, X. Wang, J. Bowers, A. Paterson, D. Lisch, and M. Freeling, “Finding and comparing syntenic regions among Arabidopsis and the outgroups papaya, poplar, and grape: CoGe with rosids.,” *Plant Physiol.*, vol. 148, no. 4, pp. 1772–81, Dec. 2008.

[87] S. M. Techtmann, A. V Lebedinsky, A. S. Colman, T. G. Sokolova, T. Woyke, L. Goodwin, and F. T. Robb, “Evidence for horizontal gene transfer of anaerobic carbon monoxide dehydrogenases.,” *Front. Microbiol.*, vol. 3, p. 132, Jan. 2012.

[88] J. Goecks, A. Nekrutenko, and J. Taylor, “Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences.,” *Genome Biol.*, vol. 11, no. 8, p. R86, Jan. 2010.

[89] S. H. Choi and E. P. Greenberg, “The C-terminal region of the Vibrio fischeri LuxR protein contains an inducer-independent lux gene activating domain.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 88, no. 24, pp. 11115–9, Dec. 1991.

[90] Z. Gelencsér, B. Galbáts, J. F. Gonzalez, K. S. Choudhary, S. Hudaiberdiev, V. Venturi, and S. Pongor, “Chromosomal arrangement of AHL-driven quorumsensing circuits in Pseudomonas.,” *ISRN Microbiol.*, p. 484176, 2012.

[91] K. R. Sakharkar, M. K. Sakharkar, C. Verma, and V. T. K. Chow, “Comparative study of overlapping genes in bacteria, with special reference to Rickettsia prowazekii and Rickettsia conorii.,” *Int. J. Syst. Evol. Microbiol.*, vol. 55, no. Pt 3, pp. 1205–9, May 2005.

[92] C. Radhakrishna Rao, “The utilization of multiple measurements in problems of biological classification,” *Journal of the Royal Statistical Society - Series B: Statistical Methodology*. John Wiley and Sons, 12-Aug-1948.

[93] H. Urbanczyk, J. C. Ast, A. J. Kaeding, J. D. Oliver, and P. V Dunlap, “Phylogenetic analysis of the incidence of lux gene horizontal transfer in Vibrionaceae.,” *J. Bacteriol.*, vol. 190, no. 10, pp. 3494–504, May 2008.

[94] H. K. Patel, P. Ferrante, S. Covaceuszach, D. Lamba, M. Scortichini, and V. Venturi, “The kiwifruit emerging pathogen Pseudomonas syringae pv. actinidiae does not produce AHLs but possesses three luxR solos.,” *PLoS One*, vol. 9, no. 1, p. e87862, Jan. 2014.

[95] S. Covaceuszach, G. Degrassi, V. Venturi, and D. Lamba, “Structural insights into a novel interkingdom signaling circuit by cartography of the ligand-binding sites of the homologous quorum sensing LuxR-family.,” *Int. J. Mol. Sci.*, vol. 14, no. 10, pp. 20578–96, Jan. 2013.

[96] K. S. Choudhary, S. Hudaiberdiev, Z. Gelencsér, B. Gonçalves Coutinho, V. Venturi, and S. Pongor, “The Organization of the Quorum Sensing luxI/R Family Genes in Burkholderia.,” *Int. J. Mol. Sci.*, vol. 14, no. 7, pp. 13727–47, Jan. 2013.

[97] M. Kleerebezem, L. E. Quadri, O. P. Kuipers, and W. M. de Vos, “Quorum sensing by peptide pheromones and two-component signal-transduction systems in Gram-positive bacteria.,” *Mol. Microbiol.*, vol. 24, no. 5, pp. 895–904, Jun. 1997.

[98] E. C. Pesci, J. B. Milbank, J. P. Pearson, S. McKnight, A. S. Kende, E. P. Greenberg, and B. H. Iglewski, “Quinolone signaling in the cell-to-cell communication system of Pseudomonas aeruginosa.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 96, no. 20, pp. 11229–34, Sep. 1999.

[99] Y. Guo, Y. Zhang, J.-L. Li, and N. Wang, “Diffusible signal factor-mediated quorum sensing plays a central role in coordinating gene

expression of Xanthomonas citri subsp. citri.," *Mol. Plant. Microbe. Interact.*, vol. 25, no. 2, pp. 165–79, Feb. 2012.

[100] H. Bi, Q. H. Christensen, Y. Feng, H. Wang, and J. E. Cronan, "The Burkholderia cenocepacia BDSF quorum sensing fatty acid is synthesized by a bifunctional crotonase homologue having both dehydratase and thioesterase activities.," *Mol. Microbiol.*, vol. 83, no. 4, pp. 840–55, Feb. 2012.

[101] J. G. Lawrence and H. Ochman, "Reconciling the many faces of lateral gene transfer," *Trends Microbiol.*, vol. 10, no. 1, pp. 1–4, Jan. 2002.

[102] W. S. Hayes and M. Borodovsky, "How to Interpret an Anonymous Bacterial Genome: Machine Learning Approach to Gene Identification," *Genome Res.*, vol. 8, no. 11, pp. 1154–1171, Nov. 1998.

[103] C. Pál, B. Papp, and M. J. Lercher, "Adaptive evolution of bacterial metabolic networks by horizontal gene transfer.," *Nat. Genet.*, vol. 37, no. 12, pp. 1372–5, Dec. 2005.