NOVEL ALIGNMENT BASED CLUSTERING ALGORITHMS FOR PAN GENOME ANALYSIS OF BACTERIA SPECIES

A Dissertation Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

By
Levent Albayrak

August 2016

NOVEL ALIGNMENT BASED CLUSTERING ALGORITHMS FOR PAN GENOME ANALYSIS OF BACTERIA SPECIES

Levent Albayrak

APPROVED:

Dr. Ioannis Pavlidis, Chairman

Dr. Zhigang Deng

Dr. Yuriy Fofanov, University of Texas Medical Branch - Galveston

Dr. Shishir Shah

Dr. Nikolaos V. Tsekos

Dean, College of Natural Sciences and Mathematics

ACKNOWLEDGEMENTS:

I wish to express sincere appreciation to the Department of Computer Science for their extended long-term support and especially to Dr. Ioannis Pavlidis and Dr. Yuriy Fofanov for their vast reserve of patience and knowledge.

I would like to show my gratitude to Dr. Zhigang Deng, Dr. Yuriy Fofanov, Dr. Shishir Shah and Dr. Nikolaos Tsekos for serving as members of the dissertation committee.

I am indebted to my numerous colleagues for their support and particularly to my teammates from the Sealy Center for Structural Biology and Molecular Biophysics at UTMB Galveston, Brenda Boyko, Kamil Khanipov, Georgiy Golovko, Mark Rojas and Maria Pimenova. I would like to present my special thanks to Dr. Michael Kosoy from the Centers for Disease Control and Prevention.

This dissertation would have never been completed without the encouragement, love and devotion of my family and friends, especially my parents Kemal Albayrak and Nedret Albayrak, my sister Damla Isbilen, her husband Saim Isbilen, my beautiful niece Defne Isbilen, and my fiancé Margarita Kim.

NOVEL ALIGNMENT BASED CLUSTERING ALGORITHMS FOR PAN GENOME ANALYSIS OF BACTERIA SPECIES

An Abstract of a Dissertation Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

By

Levent Albayrak

August 2016

ABSTRACT

Understanding the basic rules of bacterial evolution and adaptation is critical in developing new anti-bacterial drugs, the use of bacteria in biotechnology applications as well as in combating undesired consequences of bacterial presence in industrial and environmental settings such as corrosion, product spoilage, and degradation.

Accumulation of single nucleotide mutations beneficial (or neutral) for bacterial survival is a well-studied mechanism of bacterial adaptation which also reflects the time of species separation from a common ancestor (molecular clock hypothesis). The gene loss or gain due to horizontal gene transfer is another much more dynamic mechanism of bacterial adaptation. Using these mechanisms, bacteria can acquire new features such as virulence factors, locomotion ability (flagella), and heat or drug resistance.

A major functional characteristic of bacterial species is the presence of particular gene sets common to the species (core genome) together with genes that are available to individual or groups of genomes (pan genome). The technical difficulties however, lie in how one can identify the same genes or gene families in evolutionarily distant organisms:

1. Identification of a sequence-similarity threshold
2. Computational complexity of sequence clustering algorithms
3. Creation of a biologically meaningful cluster topology

In this work, we have developed methods to improve the quality and performance of gene clustering including heuristics free, novel sequence alignment algorithms able to cluster a large number of sequences significantly faster than traditional methods (a few days compared to months of computation) that permit the identification of appropriate similarity thresholds and formation of biologically meaningful cluster topology.

The developed algorithms were used to build a “functional similarity” tree of the species reflecting gene composition similarity. The performed analysis also identified co-appearance and avoidance patterns of genes in bacterial species. We have applied the proposed methods to 22 genomes from *Bartonella* spp. using 34,060 genes.

CONTENTS

Introduction

Understanding the basic rules of bacterial evolution and adaptation is critical in developing anti-bacterial drugs and using bacteria in biotechnology applications [1]–[3]. In addition, it can help develop solutions to combat undesired consequences of bacterial presence in industrial and environmental settings such as corrosion, product spoilage, and degradation. Genes, coding or non-coding, are the main functional elements of bacteria regulatory and metabolic networks, exercising of the functions required to keep the organism alive in its environment. Depending upon the species, a bacterial genome contains 1,500 to 5,000 genes [4]. Many of these genes are involved in processes that are absolutely essential for the survival of the bacteria such as replication of DNA, translation of DNA into proteins, DNA transcription, maintenance of basic cellular structure and transport or blockage of materials in and out of the cell body. The "remaining" genes carry specific functional properties that permit the bacteria to thrive in their niches. The composition of genes present in a bacteria defines the complete functional profile of the organism and its ability to survive in each given environment. Bacteria is also under selective pressure to keep its genome size limited, so one of the main mechanisms of how bacteria can change its functional repertoire is by means of selective gene loss and new gene acquisition through horizontal gene transfer [5], [6].

Horizontal gene transfer can be performed through several different mechanisms including sexual transmission of bacterial DNA, type III and type IV secretion systems, and plasmid exchange or viral (phage) activities. In fact, the fastest way for bacteria to acquire new functions and adapt to a new environment is not through the accumulation of single mutations over time, but rather, through the acquisition of genes or combinations of genes that are able to perform desired function(s) [6]. For example, genes responsible for antibiotic resistance may be transmitted from one bacteria species to another through physical means such as horizontal gene transfer, rather than through sexual reproduction and heredity.

Dynamic gene acquisition and loss mechanisms can result in members of a bacterial species sharing common genes that are essential (*core genome*) for the species' survival regardless of environmental pressure, and a flexible set of genes that are specific to the environment that are present in only some members of the species. The *pan genome* is defined as the complete pool of genes "available" (acquirable) for a given species of bacteria (including the flexible and core set of genes). Even though the *core genome* is expected to be found in every member of the species, only a small subset of the complete set of flexible genes available to the species can be seen in an individual organism due to gene gain and loss mechanisms and genome size limitations.

Gene profiles (the complete set of genes present in individual bacteria) can also be used to estimate similarities between bacteria strains based on their functional characteristics. Using gene profiles, the history of the species' evolution and adaptation can be reconstructed revealing insights into evolutionary bottlenecks, host changes, and acquisition or loss of metabolic pathways. In essence, gene profiles do not only reflect the functional changes that the species has gone through, but also its surrounding environment, host and microbial community.

Large number of studies have been conducted to compare gene profiles of organisms [7]–[10]. Several mathematical models have been developed to approximate the core and pan-genome size of bacterial species [7], [10]–[13], [8]. The following challenges, however, remain unresolved:

1- The use of arbitrarily selected nucleotide sequence-similarity thresholds in identification of homologous genes.

To date, attempts to identify homologous genes across different organisms have been based on establishing a “reasonable criteria”: such as the arbitrary sequence-similarity cutoff above which the gene sequences are considered similar. As a consequence, various studies employ different criteria to identify homologous genes which can significantly affect the outcome of the analysis [9], [8], [14].

2- The computational complexity of sequence clustering algorithms.

Clustering of thousands of gene sequences is an extremely computationally intensive task. A variety of the present clustering algorithms resort to employing greedy heuristics methods resulting in a large number of artificially fragmented clusters [15], [16].

3- The lack of biologically meaningful cluster topology.

Gene sequences “evolve” over time through single point mutations. The function of genes, however, remains the same. Centroid (representative) based clustering algorithms used in the majority of modern studies (such as UCLUST [15] and Cd-hit [16]) result in artificially “circular” clusters where the representative gene of the cluster is considered to be the only ancestor of all other genes. In the course of evolution, however, every gene can form its own lineage, so, a gene family (gene cluster) can have complex ancestor-descendant relations, which can be more appropriately described using *undirected connected graph* cluster topology.

The presented research is focused on:

1- Development of appropriate scoring rules for global alignment of bacterial gene sequences in nucleotide space.

2- Novel clustering algorithms for large scale gene clustering including:

   a. Improved global alignment algorithms.
   b. Non-centroid based clustering topology.
   c. Dynamic identification of natural clustering similarity threshold.

3- Novel methods to introduce distance between organisms based upon the similarity of their gene profiles.

As an example, developed algorithms have been applied to create functional similarity profiles and create a novel mathematical model to describe the core/pan genome composition of *Bartonella* spp.

1. Clustering

Gene sequences change over time through single point mutations. The function of genes, however, remains the same. Identification and comparison of gene composition across different bacterial species requires clustering of thousands of gene sequences. To date, a variety of algorithms have been proposed to perform this task [15], [16]. Computational complexity and large memory requirement, however, result in the development of various heuristics compromising clustering quality.

Each gene clustering approach contains two steps: pairwise sequence alignment and the use of this information for the formation of clusters. The performance of every clustering algorithm primarily depends on how many pairwise alignments it has to perform, consequently, many heuristics focus on ways to reduce the number of alignments. For example, UCLUST [15] and Cd-hit [16] use predefined sequence-similarity thresholds where each sequence is assigned to an existing cluster and excluded from further analysis when the distance between the sequence and cluster representative sequence (centroid of the cluster) is below a threshold. The "random" selection of cluster representative sequences where each sequence not assigned to an existing cluster immediately becomes a representative sequence of a new cluster which can lead to a *false positive* new cluster, resulting in homologous genes to appear in different clusters. Another disadvantage of predefined clustering thresholds is that clustering has to be re-done for each different threshold. This approach also causes clusters to be artificially "circular" where the representative sequence of the cluster is considered to be the only evolutionary ancestor of all other genes in the cluster. In the course of evolution, however, any gene can form its own lineage, therefore, a gene family can have complex ancestor-descendant relations, which can be described better using undirected connected graph cluster topology (Figure 1.1).

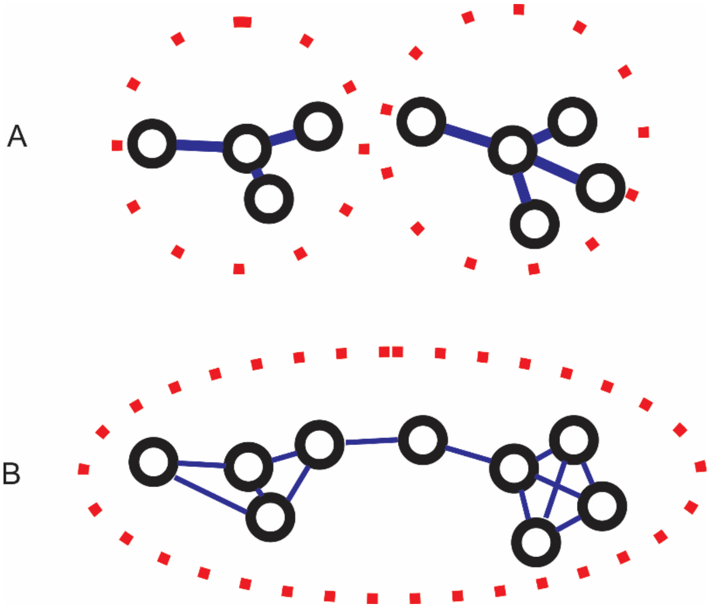


Figure 1.1. Centroid (A) and undirected connected graph (B) cluster topology

1.1 Reducing the Number of Pairwise Alignments Using Gene Length Differences

A typical example of the complexity and scale of clustering can be illustrated by the task of identifying homologous genes in 22 *Bartonella* spp. reference genomes. The significant scientific and practical interest in this bacteria causes cat scratch disease [17], [18], trench fever [18]–[21], and Carrion's disease [18], [22] in humans. Transmission vectors [23] for this organism include ticks, fleas, sand flies, and mosquitoes which can move this bacteria among various mammalian hosts [24] such as rats, bats, canines etc. With an average genome size of 1.5 million nucleotides, this bacteria has between 1,200 and 1,900 genes. The gene length varies from 100 to 16,000 nucleotides (Figure 1.2).

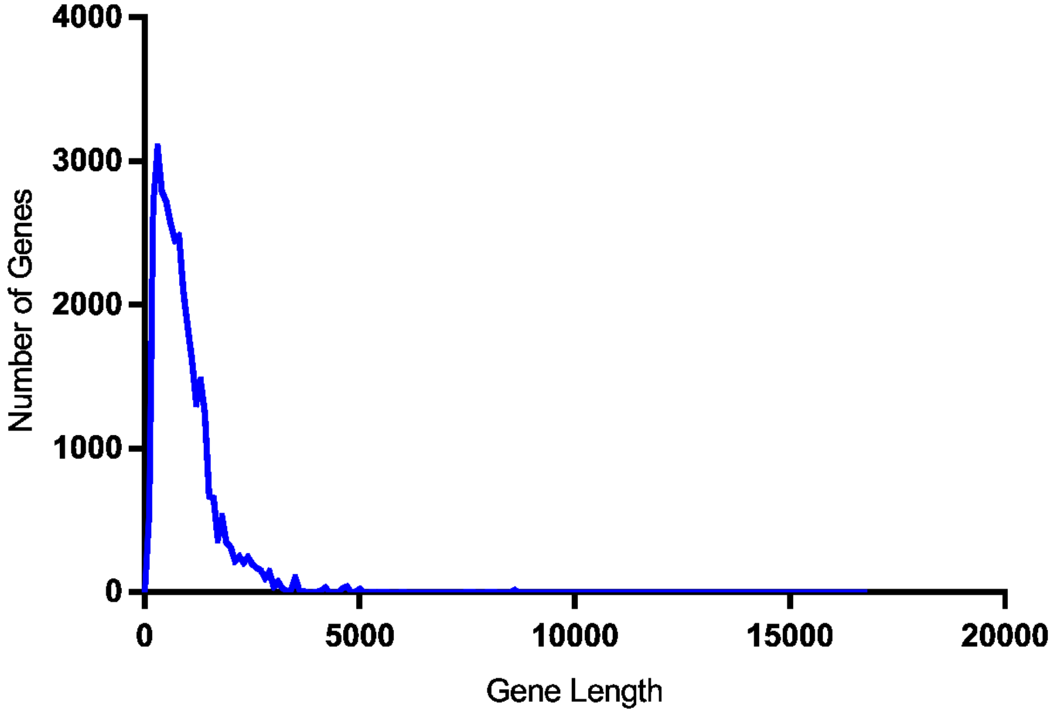


Figure 1.2 Gene length distribution for 34,060 genes from 22 Bartonella genomes.

Using the naïve clustering approach to identify common genes in these 22 genomes requires performing ~578 million pairwise alignments for approximately 34,000 genes. The only way to improve performance of the clustering is to avoid unnecessary alignments.

The greedy approach used in UCLUST [15] and Cd-hit [16] algorithms tries to exclude sequences from future consideration as soon as they become a certain distance from the existing clusters. Another method is to compare the subsequence composition of the sequences using, for example, different order Markov models [25] to identify significantly different sequences not expected to produce reasonable alignment scores. Unfortunately, such approaches do not consider the different means by which alignment scores can be calculated: difference in penalties of gaps and mismatches cannot be accommodated by Markov models.

To address these challenges, we propose to take advantage of the significant variation in gene lengths. This property of gene sequences allows determination of whether the alignment score of a pair of gene sequences meets the minimum alignment score threshold by simply comparing their lengths. Assuming that in the best case scenario, the shorter sequence has to be a complete subsequence of the longer sequence (with no mismatches), the length difference can be used to predict the maximum possible alignment score between these sequences without comparing them (Formula 1.1):

Formula 1.1. Maximum possible alignment score between two sequences

$$maximum\ possible\ score = \frac{\min\left(\left||s_1|\right|, \left||s_2|\right|\right) * score_{match} + \left|\left||s_1|\right| - \left||s_2|\right|\right| * score_{gap}}{\max\left(\left||s_1|\right|, \left||s_2|\right|\right) * score_{match}}$$

When the maximum possible alignment score cannot meet the threshold requirement, the alignment can be avoided.

Sorting gene sequences by length can be done using the *most significant digit radix sort* [26] algorithm ($O(N)$ time complexity, where N is the number of sequences to be clustered), so if the pairwise alignment tasks begin from the longest sequences, it can be terminated as soon as the sequence length difference reaches the threshold limit (figure 1.3). This approach reduces the clustering complexity from $O(N^2)$ to $O(N^{\log_N k})$ where k is the average number of genes within the length (maximum possible score above threshold) range of each gene.

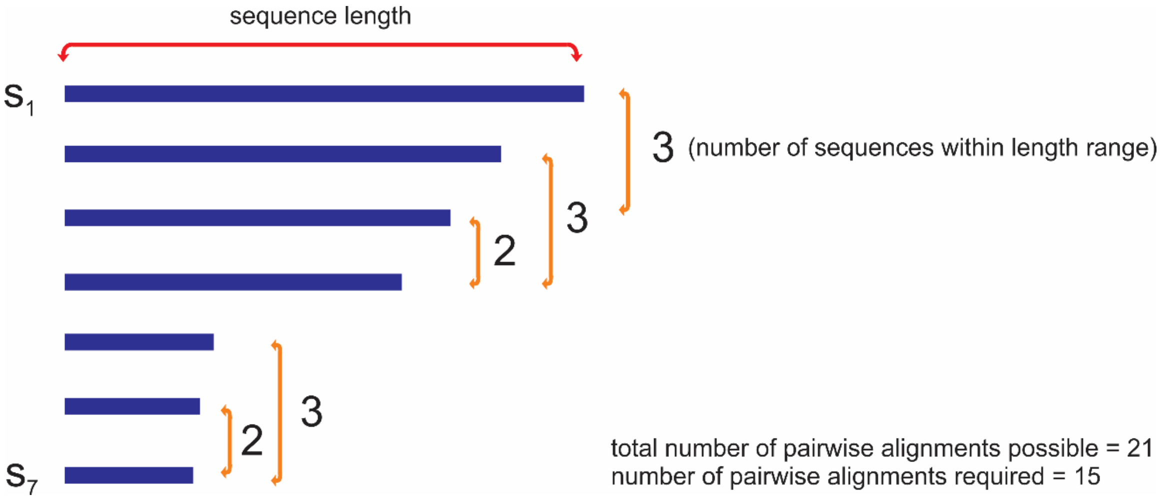

sequence length
S_1
S_7
3 (number of sequences within length range)
3
2
3
2
total number of pairwise alignments possible = 21
number of pairwise alignments required = 15

Figure 1.3. Using sorting and sequence length differences to avoid unnecessary pairwise alignment during clustering.

Figure 1.4 shows an example that compares the naïve approach to the sequence length difference-based termination approach. As one can see, for both real and simulated gene sequences, the sequence length difference-based termination approach can produce about a 9-fold improvement in performance.

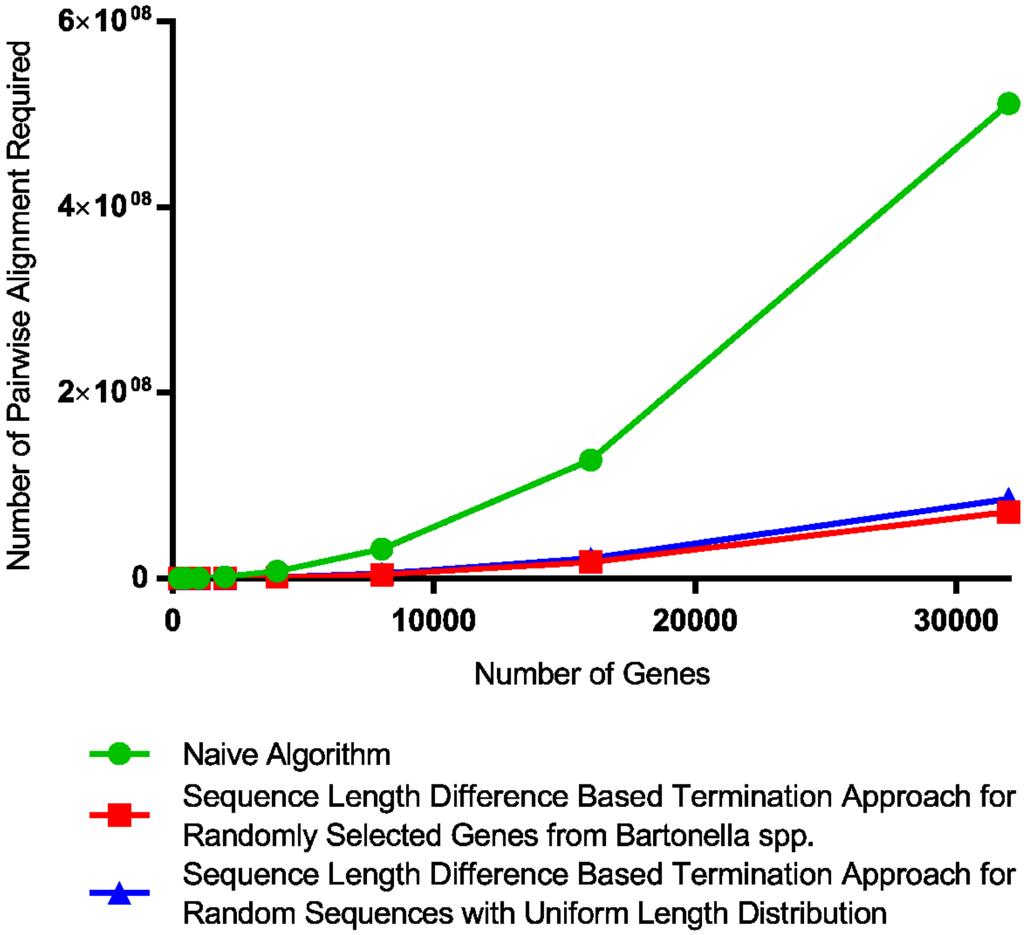


Figure 1.4. Number of pairwise alignments required for clustering.

Many clustering methods in use today employ heuristic clustering methods due to the computational complexity of an exhaustive sequence comparison [27]. The proposed approach can significantly improve the performance and computational complexity of clustering large sets of sequences without introducing missing cases. The ability to perform exhaustive pairwise comparisons makes it possible to employ clustering methods that can create biologically-meaningful cluster topologies not possible with heuristics based approaches.

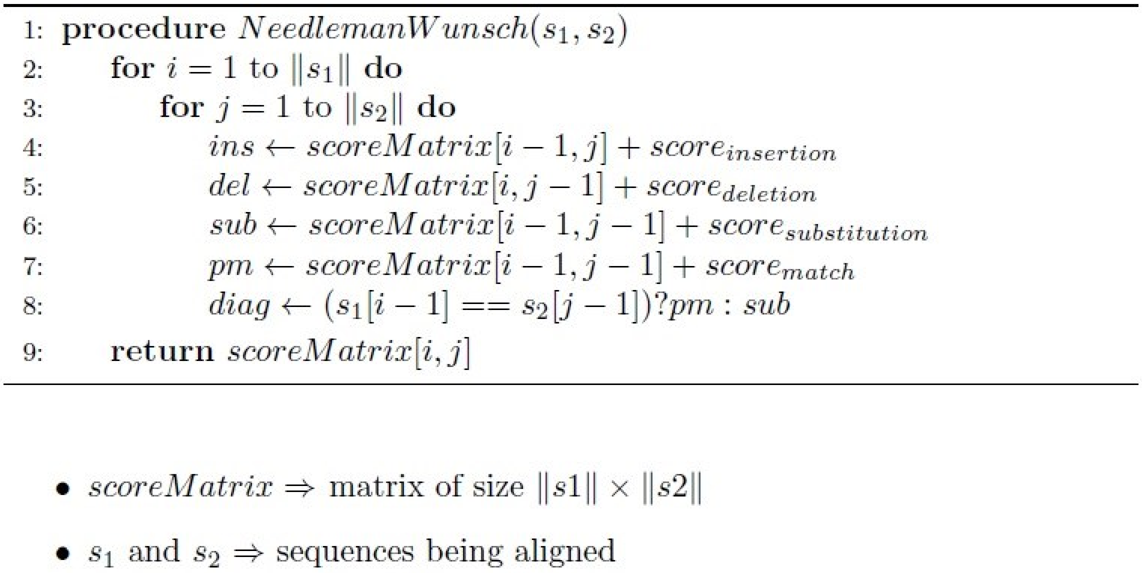
2. Sequence Alignment Algorithm

Described in the previous chapter, optimal cluster-formation strategy focuses on minimizing the number of pairwise sequence comparisons (alignments) to be performed during the clustering process. Alignment itself however, is the most time consuming part of the gene clustering task. There are two major types of pairwise alignment algorithms: global alignment [28], when two sequences must be aligned completely from the beginning to the end, and local alignment [29] focused on identification of similar regions between the sequences under investigation. Keeping in mind that to be homologous, two genes must be similar across the entire sequence, gene clustering algorithms require employment of global alignment.

The global alignment algorithms are widely used to solve the so-called “optimal matching problem” that maximizes the similarity score calculated using pairwise correspondence between nucleotides and artificially introduced gaps across two sequences. The output of global alignment algorithms is the overall *alignment score* value and if necessary, pairwise correspondence information, usually called “*alignment*”.

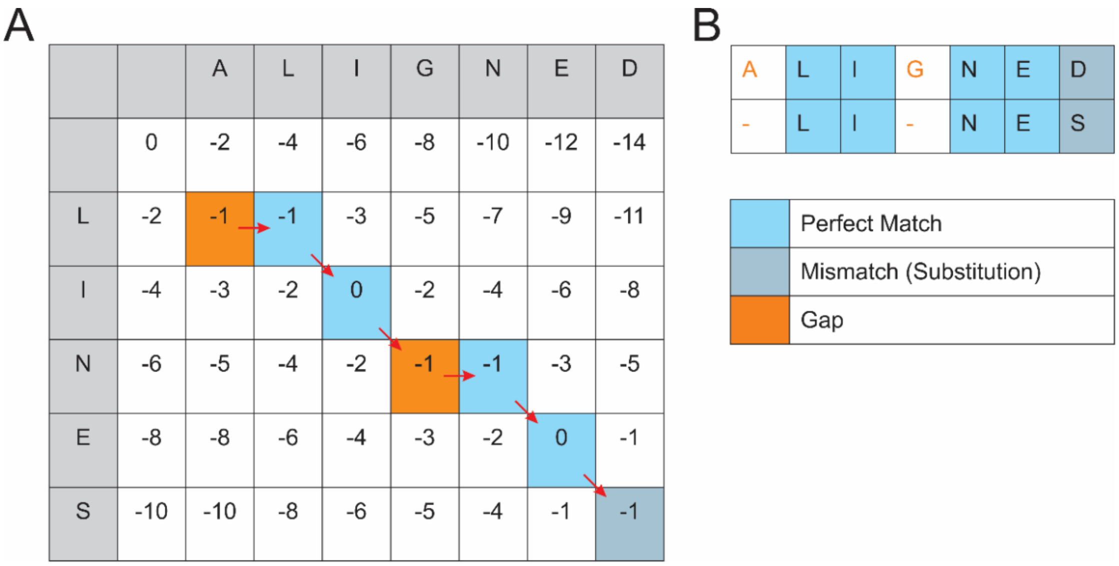
The *Needleman-Wunsch* global alignment algorithm (Algorithm 2.1, Figure 2.1) developed by Saul B. Needleman and Christian D. Wunsch and published in 1970 [28], is the first choice of precise global alignment. The original implementation of this dynamic programming algorithm has $O(m^2)$ time as well as space complexity. Numerous variations of this algorithm have been developed to reduce its computational complexity, such as, memory efficient variants (primarily based on Hirschberg's algorithm [30]) and time efficient variants such as a partial calculation of the score-matrix [31], and branch-and-bound-tree based methods [32].

Algorithm 2.1. Needleman-Wunsch:



```
procedure $NeedlemanWunsch(s_1, s_2)$
    for $i = 1$ to $\|s_1\|$ do
        for $j = 1$ to $\|s_2\|$ do
            $ins \leftarrow scoreMatrix[i-1, j] + score_{insertion}$
            $del \leftarrow scoreMatrix[i, j-1] + score_{deletion}$
            $sub \leftarrow scoreMatrix[i-1, j-1] + score_{substitution}$
            $pm \leftarrow scoreMatrix[i-1, j-1] + score_{match}$
            $diag \leftarrow (s_1[i-1] == s_2[j-1])?pm : sub$
    return $scoreMatrix[i, j]$
```

- $scoreMatrix \Rightarrow$ matrix of size $\|s1\| \times \|s2\|$
- s_1 and $s_2 \Rightarrow$ sequences being aligned



		A	L	I	G	N	E	D
	0	-2	-4	-6	-8	-10	-12	-14
L	-2	-1	-1	-3	-5	-7	-9	-11
I	-4	-3	-2	0	-2	-4	-6	-8
N	-6	-5	-4	-2	-1	-1	-3	-5
E	-8	-8	-6	-4	-3	-2	0	-1
S	-10	-10	-8	-6	-5	-4	-1	-1

A	L	I	G	N	E	D
-	L	I	-	N	E	S

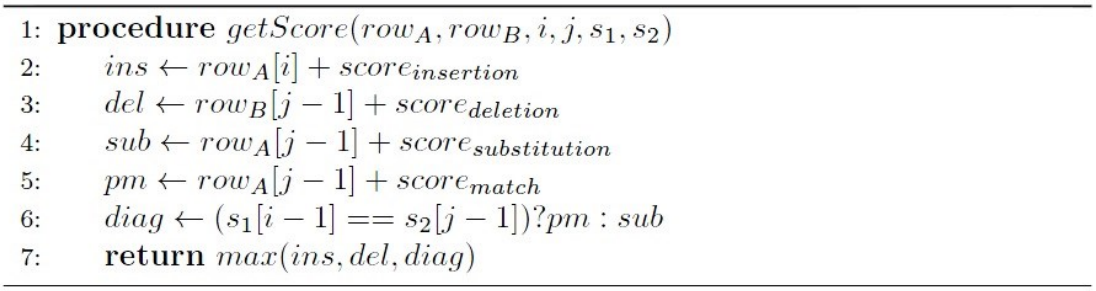
Figure 2.1. Needleman-Wunsch algorithm. A) Score-matrix (scoring rules: perfect match = 1, substitution = -1, gap = -2) B) Alignment

These efforts have been made to improve this algorithm without imposing additional limitations on the alignment process. However, taking into consideration specifics which may appear in different sequence alignment tasks, can be used to improve the case specific performance of the algorithm. For example, in the gene clustering approach, one is not required to generate an alignment traceback to be used in the downstream cluster formation. The alignment score for dissimilar sequences is also not needed to form clusters.

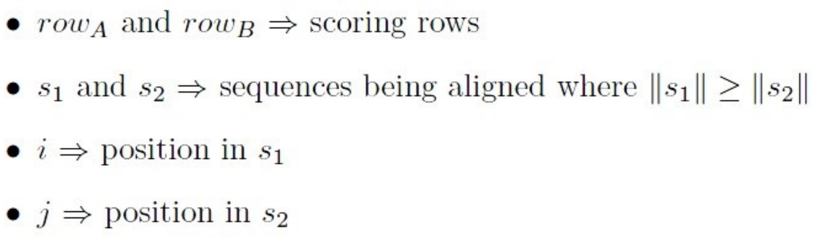
2.1 Small memory footprint: Hirschberg’s Algorithm

The original implementation of the Needleman-Wunsch algorithm requires an m^2 size matrix [28] (where m is the sequence length) to store partial alignment scores. This matrix is also used to traceback optimal alignment(s) in the back-tracking step of the algorithm. Hirschberg’s algorithm provides memory reduction ($O(m)$) and performance improvement, especially if the traceback step is not required and only the final alignment score is needed. The basic idea of the Hirschberg’s algorithm [30] is to only keep in memory the two rows of the Needleman-Wunsch score-matrix and (in our implementation) swap the rows using pointers without reallocating them in memory. Another advantage of this algorithm is that multiple pairwise alignments can be done in parallel taking advantage of a small memory footprint.

Algorithm 2.2. getScore utility function:

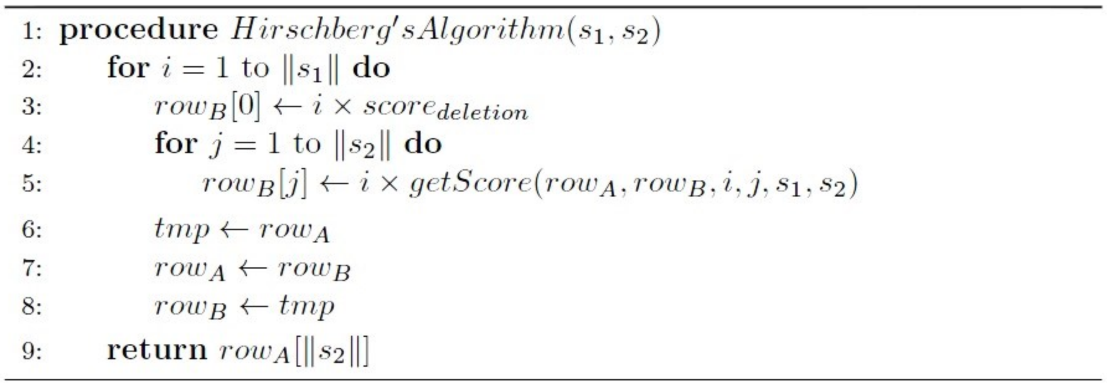


```
**procedure** $getScore(row_A, row_B, i, j, s_1, s_2)$
    $ins \leftarrow row_A[i] + score_{insertion}$
    $del \leftarrow row_B[j-1] + score_{deletion}$
    $sub \leftarrow row_A[j-1] + score_{substitution}$
    $pm \leftarrow row_A[j-1] + score_{match}$
    $diag \leftarrow (s_1[i-1] == s_2[j-1])?pm : sub$
    **return** $max(ins, del, diag)$
```

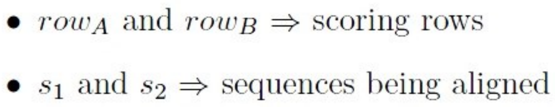


- row_A and $row_B \Rightarrow$ scoring rows
- s_1 and $s_2 \Rightarrow$ sequences being aligned where $\|s_1\| \geq \|s_2\|$
- $i \Rightarrow$ position in s_1
- $j \Rightarrow$ position in s_2

Algorithm 2.3. Hirschberg’s algorithm:



```
procedure $Hirschberg'sAlgorithm(s_1, s_2)$
    for $i = 1$ to $\|s_1\|$ do
        $row_B[0] \leftarrow i \times score_{deletion}$
        for $j = 1$ to $\|s_2\|$ do
            $row_B[j] \leftarrow i \times getScore(row_A, row_B, i, j, s_1, s_2)$
        $tmp \leftarrow row_A$
        $row_A \leftarrow row_B$
        $row_B \leftarrow tmp$
    return $row_A[\|s_2\|]$
```



- row_A and row_B ⇒ scoring rows
- s_1 and s_2 ⇒ sequences being aligned

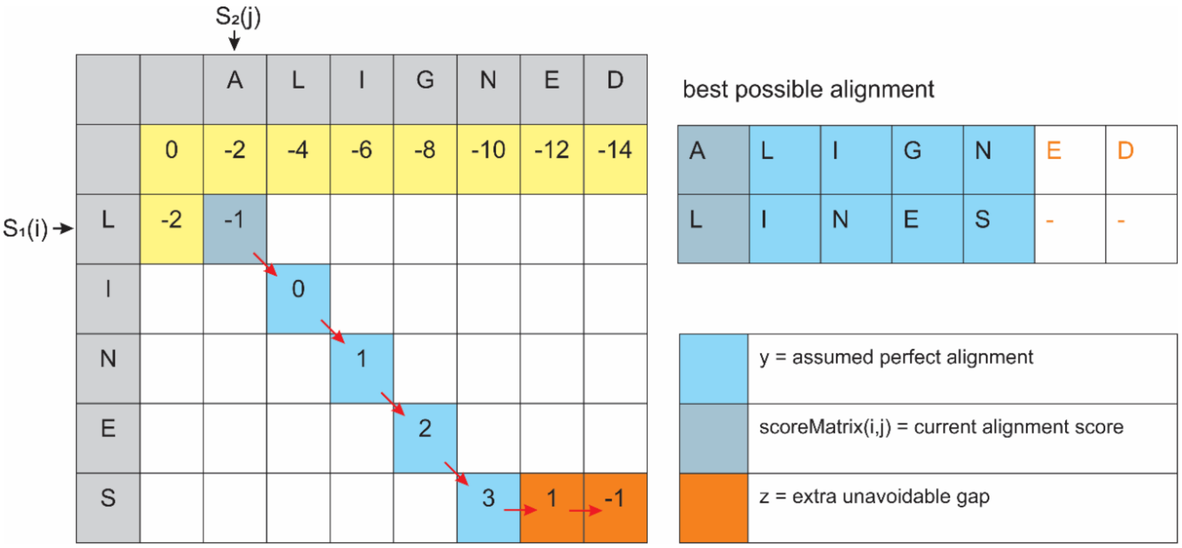
2.2 Time efficiency improvement: Early Alignment Termination

Considering that a majority of the pairwise alignments in gene clustering tasks will not produce significant alignment scores to be used in cluster formation, early identification and removal of cases that do not produce acceptable scores can significantly improve the overall performance of the clustering algorithm. The following algorithms have been designed to take advantage of such specific properties of the gene clustering task.

2.2.1 Early Alignment Termination Using Diagonal-Extension

The basic idea of the diagonal-extension based alignment termination is that calculation of the score-matrix values can be terminated once it is determined that the final alignment score can no longer meet the minimum-required threshold (Formula 2.1). This can be achieved by representing the further steps of alignment as a perfect match diagonal-extension (‘y’ in Figure 2.2) of the existing alignment (‘$scoreMatrix(i,j)$’ in Figure 2.2)). Depending on where the diagonal meets the edge of the scoring matrix (based on the origin of the diagonal’s location in the score-matrix), the differences in length between the two gene sequences and the location of the cell could result in unavoidable insertions or deletions (‘z’ in Figure 2.1). Combining the scores obtained from the diagonal’s perfect matches and the unavoidable gaps results in the best possible score that can be produced from a given location in the score-matrix (Algorithm 2.4).

The best possible score calculation for a given location in s_1 (Formula 2.1) is illustrated in Figure 2.2.



		A	L	I	G	N	E	D
	0	-2	-4	-6	-8	-10	-12	-14
L	-2	-1						
I			0					
N				1				
E					2			
S						3	1	-1

best possible alignment

A	L	I	G	N	E	D
L	I	N	E	S	-	-

🟦	y = assumed perfect alignment
■	scoreMatrix(i,j) = current alignment score
🟧	z = extra unavoidable gap

Figure 2.2. Illustration of best possible score achievable from a given location in $s_1(i)$ *and* $s_2(j)$ *.*

Formula 2.1. Best possible score calculation at position i, j, where $\left||s_1|\right| \geq \left||s_2|\right|$:

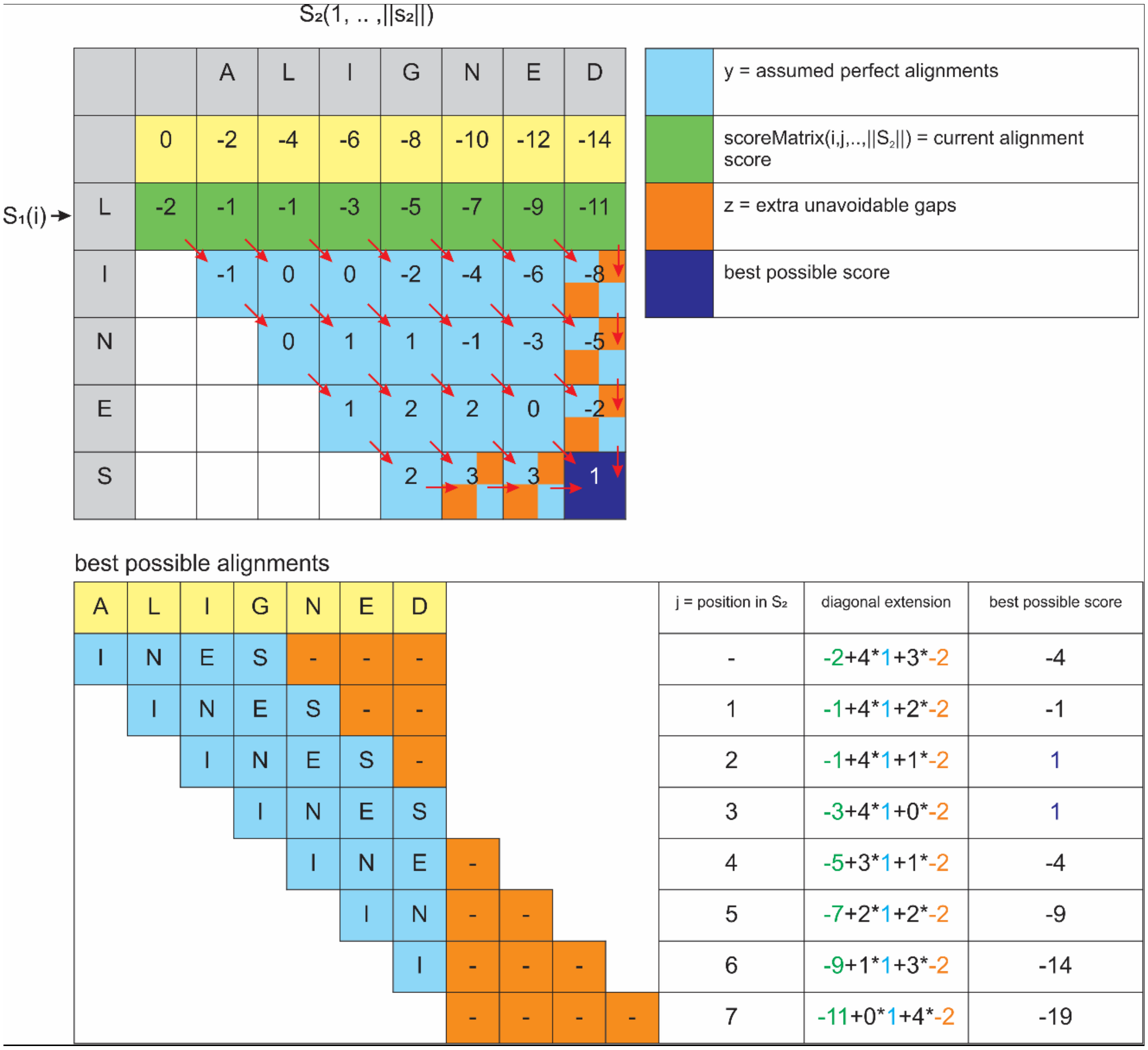
$$best\ possible\ score(i,j) = \frac{scoreMatrix(i,j) + y * score_{match} + z * score_{gap}}{||s_1|| * score_{match}}$$

where:

$$y = \min\left(\left||s_1|\right| - i, \left||s_2|\right| - j\right)$$

$$z = \max\left(\left||s_1|\right| - i, \left||s_2|\right| - j\right) - y \ \Rightarrow \left|\left(\left||s_1|\right| - i\right) - \left(\left||s_2|\right| - j\right)\right|$$

Similarly, the best score achievable at a given row (one position in s_1 for all positions in s_2) is the maximum of the best possible scores at each position in the row (Figure 2.3 and Formula 2.2)



		A	L	I	G	N	E	D
	0	-2	-4	-6	-8	-10	-12	-14
L	-2	-1	-1	-3	-5	-7	-9	-11
I		-1	0	0	-2	-4	-6	-8
N			0	1	1	-1	-3	-5
E				1	2	2	0	-2
S					2	3	3	1

🟦	y = assumed perfect alignments
🟩	scoreMatrix(i,j,..,$\|\|S_2\|\|$) = current alignment score
🟧	z = extra unavoidable gaps
■	best possible score

best possible alignments

A	L	I	G	N	E	D					j = position in S₂	diagonal extension	best possible score
I	N	E	S	-	-	-					-	-2+4*1+3*-2	-4
	I	N	E	S	-	-					1	-1+4*1+2*-2	-1
		I	N	E	S	-					2	-1+4*1+1*-2	1
			I	N	E	S					3	-3+4*1+0*-2	1
				I	N	E	-				4	-5+3*1+1*-2	-4
					I	N	-	-			5	-7+2*1+2*-2	-9
						I	-	-	-		6	-9+1*1+3*-2	-14
							-	-	-	-	7	-11+0*1+4*-2	-19

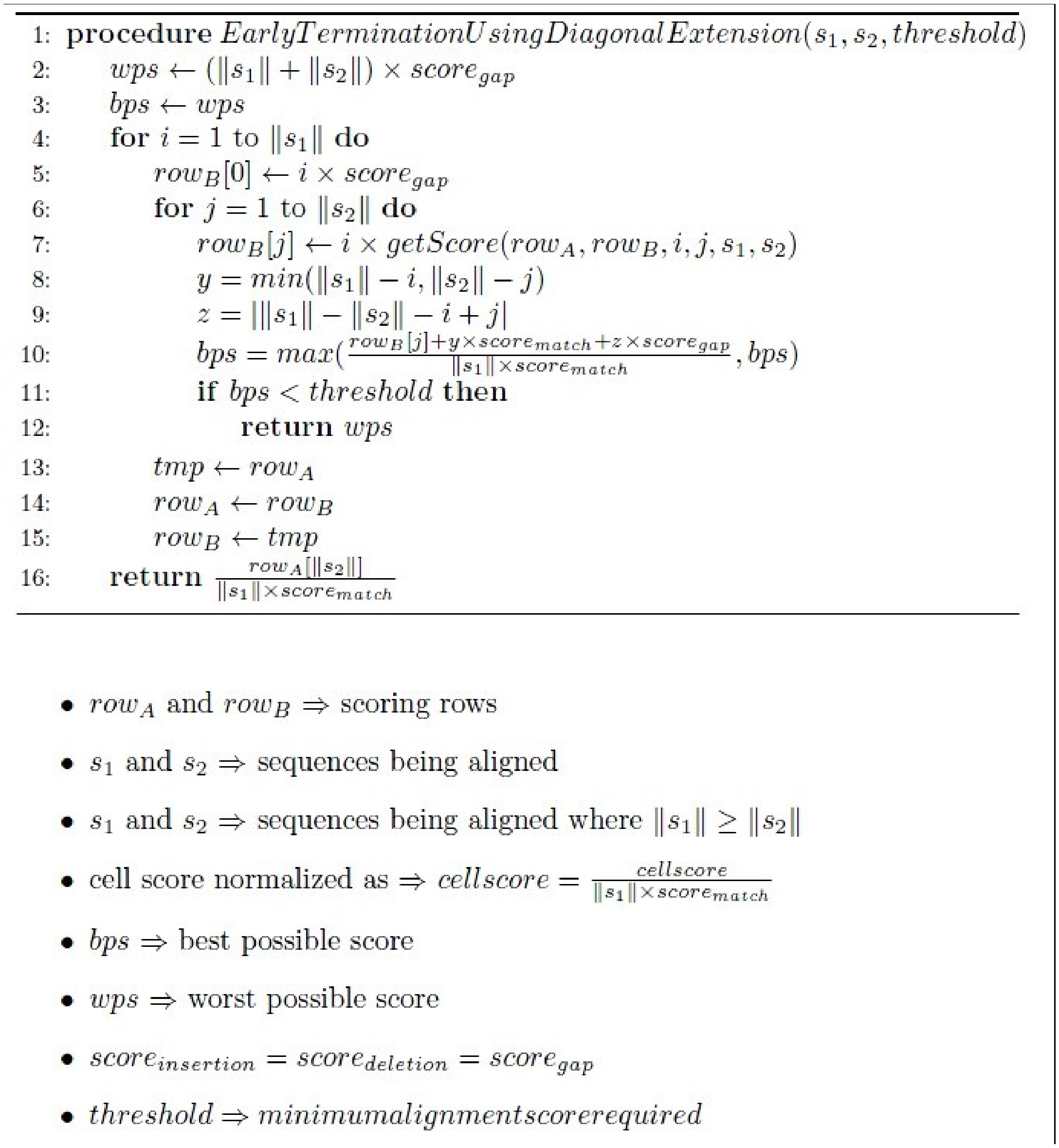
Figure 2.3. Illustration of the best score achievable from a row – diagonal-extension for early-termination.

Formula 2.2. Best possible score achievable at row i

$$best\ possible\ score(i) = max_{j=0}^{||s_2||}\left(\frac{scoreMatrix(i,j)+y*score_{match}+z*score_{gap}}{||s_1||*score_{match}}\right)$$

The proposed algorithm for early-termination can be defined as follows:

Algorithm 2.4. Modified Hirschberg algorithm for early alignment termination using diagonal-extension:

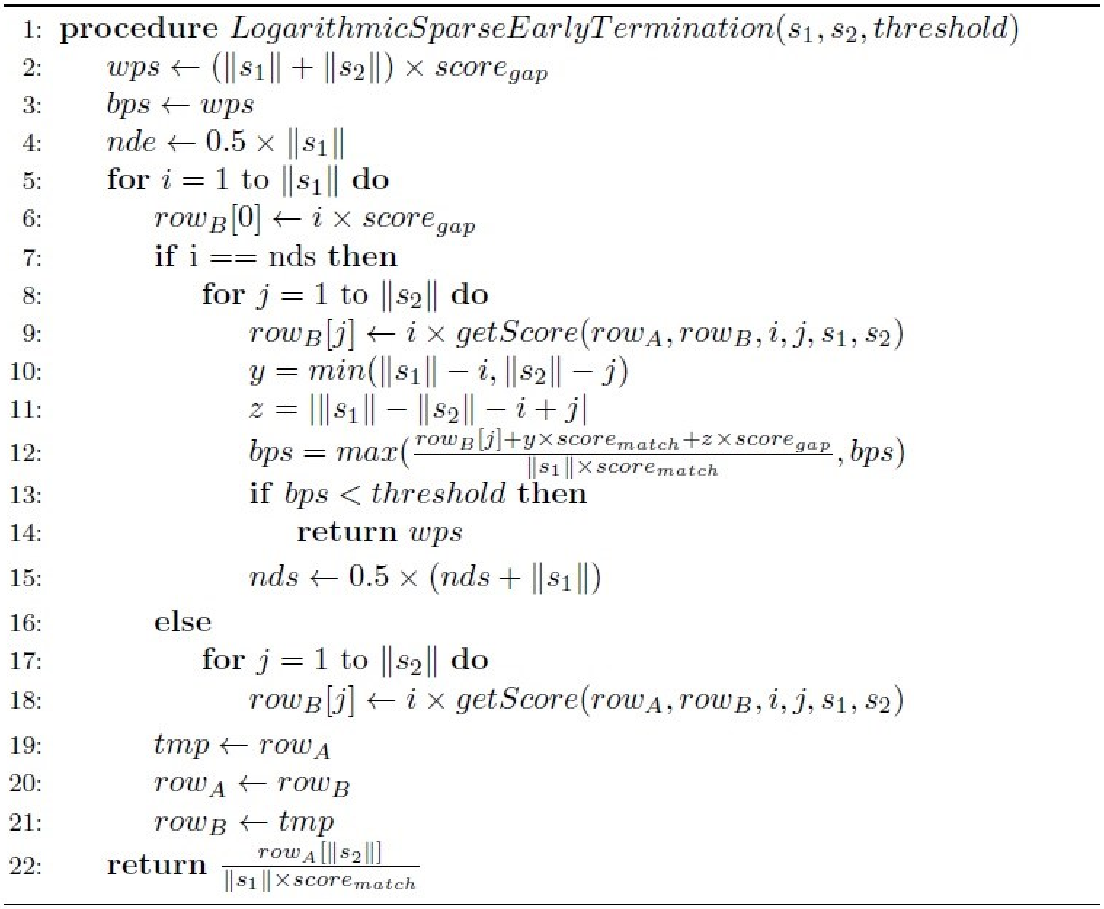


```
procedure $EarlyTerminationUsingDiagonalExtension(s_1, s_2, threshold)$
    $wps \leftarrow (\|s_1\| + \|s_2\|) \times score_{gap}$
    $bps \leftarrow wps$
    for $i = 1$ to $\|s_1\|$ do
        $row_B[0] \leftarrow i \times score_{gap}$
        for $j = 1$ to $\|s_2\|$ do
            $row_B[j] \leftarrow i \times getScore(row_A, row_B, i, j, s_1, s_2)$
            $y = min(\|s_1\| - i, \|s_2\| - j)$
            $z = |\|s_1\| - \|s_2\| - i + j|$
            $bps = max(\frac{row_B[j]+y\times score_{match}+z\times score_{gap}}{\|s_1\|\times score_{match}}, bps)$
            if $bps < threshold$ then
                return $wps$
        $tmp \leftarrow row_A$
        $row_A \leftarrow row_B$
        $row_B \leftarrow tmp$
    return $\frac{row_A[\|s_2\|]}{\|s_1\|\times score_{match}}$
```

- row_A and row_B ⇒ scoring rows
- s_1 and s_2 ⇒ sequences being aligned
- s_1 and s_2 ⇒ sequences being aligned where $\|s_1\| \geq \|s_2\|$
- cell score normalized as ⇒ $cellscore = \frac{cellscore}{\|s_1\|\times score_{match}}$
- bps ⇒ best possible score
- wps ⇒ worst possible score
- $score_{insertion} = score_{deletion} = score_{gap}$
- $threshold \Rightarrow minimumalignmentscorerequired$

The calculation of the best possible score for each position in the score-matrix can add additional overhead to the performance of the alignment algorithm. This is especially true if two sequences are very similar. In order to minimize the number of unnecessary calculations of a best possible score, we propose to use a *sparse approach* where the best possible score is calculated only for $\log(m)$ locations (Algorithm 2.6).

Algorithm 2.5. Logarithmic-sparse early-termination:



```
procedure $LogarithmicSparseEarlyTermination(s_1, s_2, threshold)$
    $wps \leftarrow (\|s_1\| + \|s_2\|) \times score_{gap}$
    $bps \leftarrow wps$
    $nde \leftarrow 0.5 \times \|s_1\|$
    for $i = 1$ to $\|s_1\|$ do
        $row_B[0] \leftarrow i \times score_{gap}$
        if i == nds then
            for $j = 1$ to $\|s_2\|$ do
                $row_B[j] \leftarrow i \times getScore(row_A, row_B, i, j, s_1, s_2)$
                $y = min(\|s_1\| - i, \|s_2\| - j)$
                $z = |\|s_1\| - \|s_2\| - i + j|$
                $bps = max(\frac{row_B[j] + y \times score_{match} + z \times score_{gap}}{\|s_1\| \times score_{match}}, bps)$
                if $bps < threshold$ then
                    return $wps$
                $nds \leftarrow 0.5 \times (nds + \|s_1\|)$
        else
            for $j = 1$ to $\|s_2\|$ do
                $row_B[j] \leftarrow i \times getScore(row_A, row_B, i, j, s_1, s_2)$
        $tmp \leftarrow row_A$
        $row_A \leftarrow row_B$
        $row_B \leftarrow tmp$
    return $\frac{row_A[\|s_2\|]}{\|s_1\| \times score_{match}}$
```

2.3 Performance test: Reduction in number of calculations by early alignment termination approach

The number of cells in the score-matrix for which the score value must be calculated can be used as an indicator of performance (improvement) of the alignment algorithm. When the original Needleman-Wunsch requires m^2 score values to be calculated, the early-termination approach will leave certain locations in the score-matrix “empty”. To estimate the reduction of score calculations by the early-termination diagonal-extension approach, we used a subset of 1,000 randomly selected sequences of *Bartonella* spp. genes with the same length range. The average percentage of calculations avoided by using early-termination and diagonal-extension was found to be 16.55% ± 14.16%, while the logarithmic-sparse approach achieved 12.02% ± 10.76% decrease in calculations (see figure 2.4 for more details).

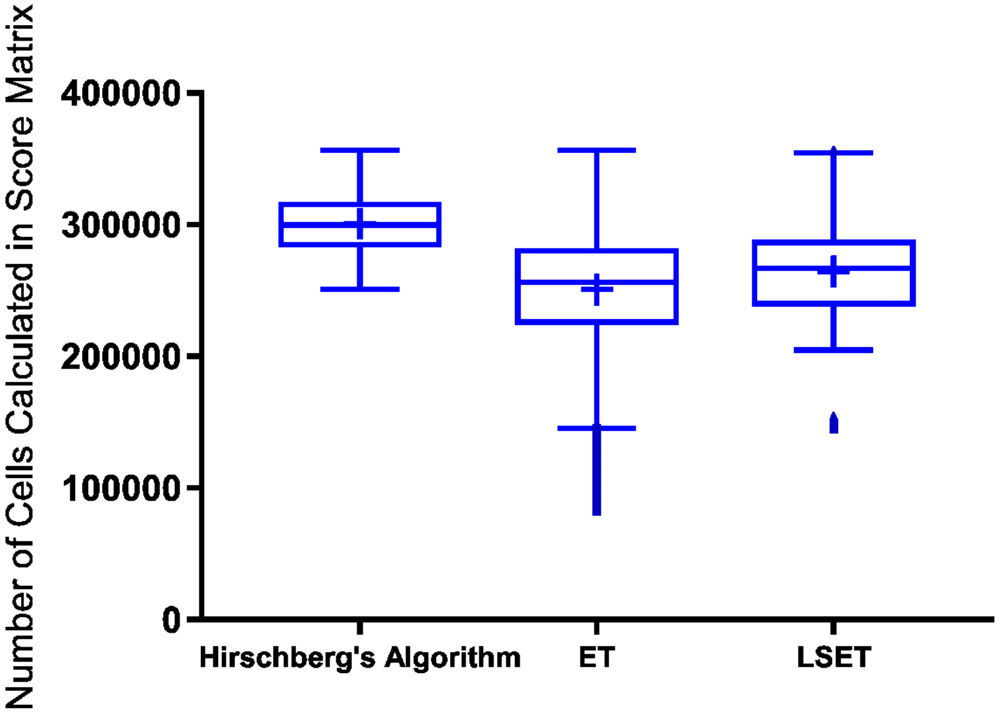


Figure 2.4. Performance comparison for Hirschberg's algorithm, early-termination (ET), and logarithmic-sparse early-termination (LSET)

2.3.1 Performance test: Logarithmic-sparse early-termination running time

In order to estimate the real-time gain of the logarithmic-sparse early-termination approach, 2,500 genes (500 nucleotides long) from *Bartonella* spp. were chosen for bootstrap testing. For each subset of gene sequences (Figure 2.5) the alignment experiment has been performed five times. The average run time gain was found to be ~10% better than the original Hirschberg's algorithm.

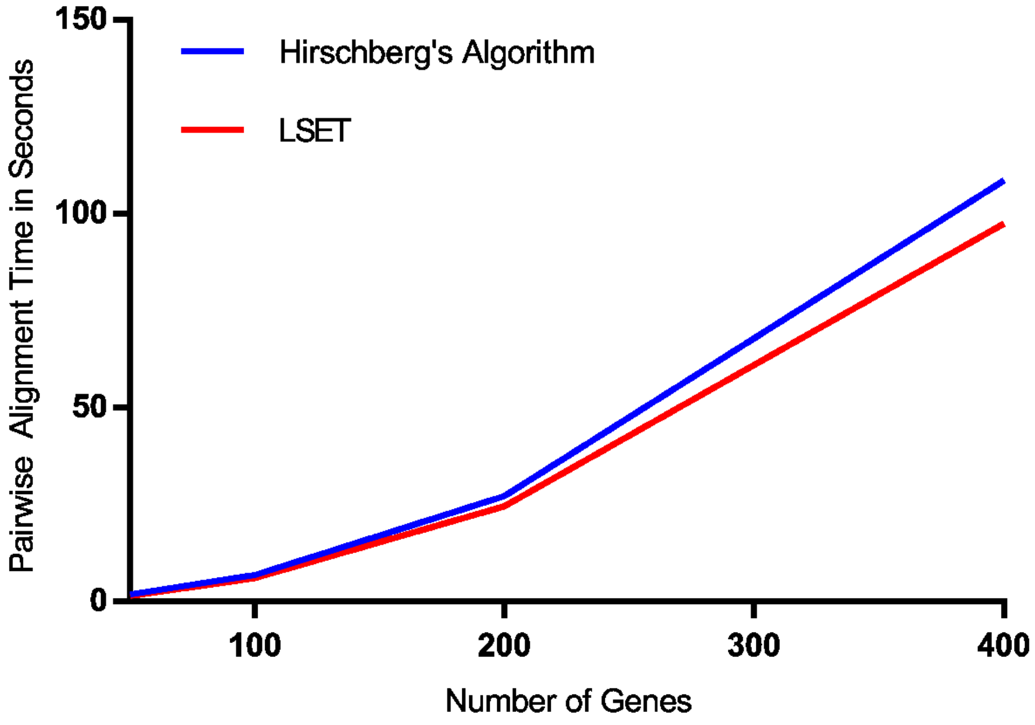


Figure 2.5. Hirschberg's algorithm and logarithmic-sparse early-termination (LSET) time comparison.

The comparison of the diagonal-extension based early-termination approach reveals that minor additions to the original Hirschberg's algorithm improved the overall performance of alignment up to ~10% which in a "real life scenario" can result in saving hours of computations.

2.4 Early Alignment Termination Using "Corridor Restriction"

The basic idea of the *corridor restriction* based alignment termination is that in order to produce a sufficient alignment score, the total number of mismatches (insertions, deletions, and substitutions) between two sequences must be limited.

The maximum number of mismatches depends on the type of mismatch, the scoring rules, and penalties for different types of mismatches and rewards for perfect matches. These requirements for the alignment permit the identification of locations in the score-matrix that cannot be part of the final alignment. They are excluded from consideration.

To identify locations where calculations can be avoided, the scoring rules require consideration. Deviation from the diagonal is caused by gaps only (perfect matches and substitutions may not make the alignment deviate from the diagonal), and can be estimated using the following formula (Formula 2.3) as seen in Figure 2.6.

Formula 2.3. Calculation of allowed gap (deviation from diagonal):

$$agap = \ allowed\ gap$$

$$t = score\ threshold$$

$$\frac{\left(\left||s_1|\right| - \left||s_2|\right| + 2 * agap\right) * score_{gap} + \left(\left||s_2|\right| - agap\right) * score_{match}}{\left||s_1|\right| * score_{match}} \geq t$$

$$agap = \frac{t * \left||s_1|\right| * score_{match} - \left||s_2|\right| * score_{match} - \left(\left||s_1|\right| - \left||s_2|\right|\right) * score_{gap}}{2 * score_{gap} - score_{match}}$$

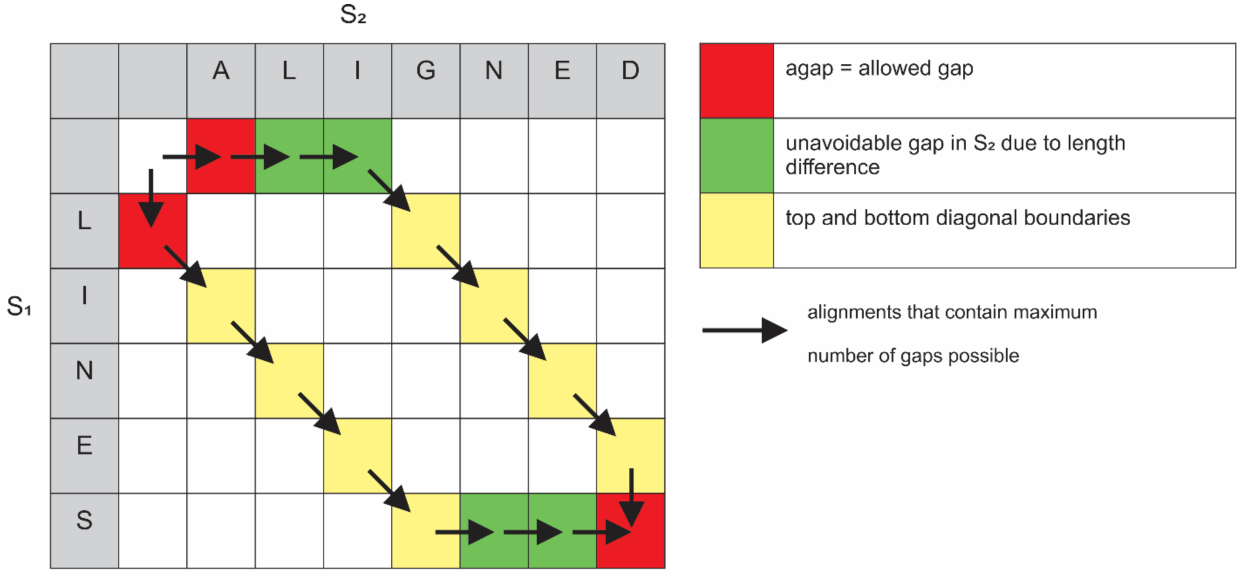
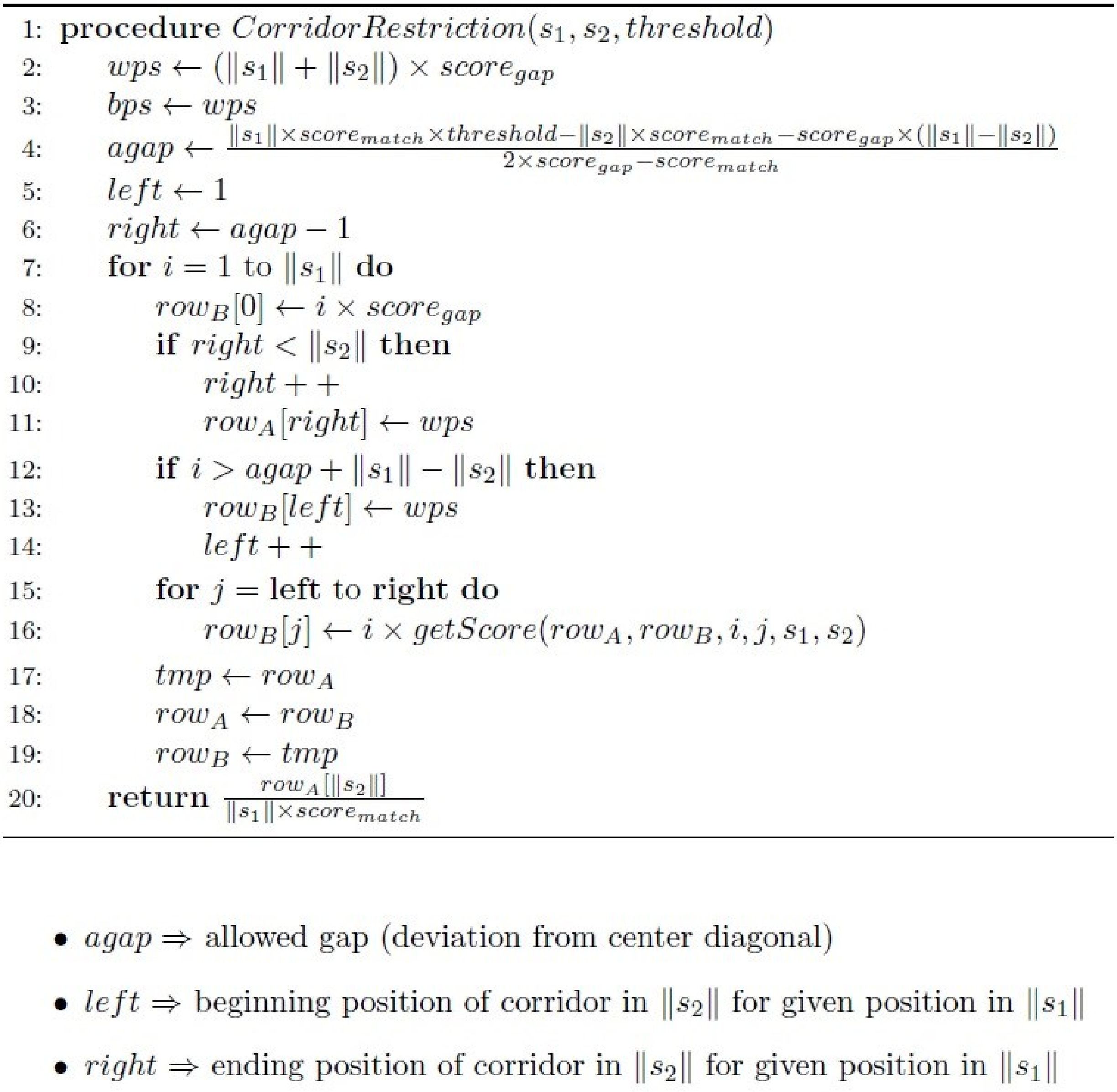

S₂
A
L
I
G
N
E
D
S₁
L
I
N
E
S
agap = allowed gap
unavoidable gap in S₂ due to length difference
top and bottom diagonal boundaries
alignments that contain maximum number of gaps possible

Figure 2.6. Illustration of allowed gap (abbreviated as agap).

When the corridor restriction reduces the amount of calculations in the score-matrix, the best results can be achieved if it is used in conjunction with the early-termination approach. When the algorithm performs the score calculations in the corridor from top left to bottom right corner of the score-matrix (Figure 2.7), one can simultaneously monitor the best possible alignment values across the partial row and terminate the alignment as soon as the best possible alignment score drops below the required minimum threshold (Figure 2.7). Figure 2.7 illustrates the basic idea of corridor restriction and early-termination.

Algorithm 2.6. Corridor restriction algorithm:

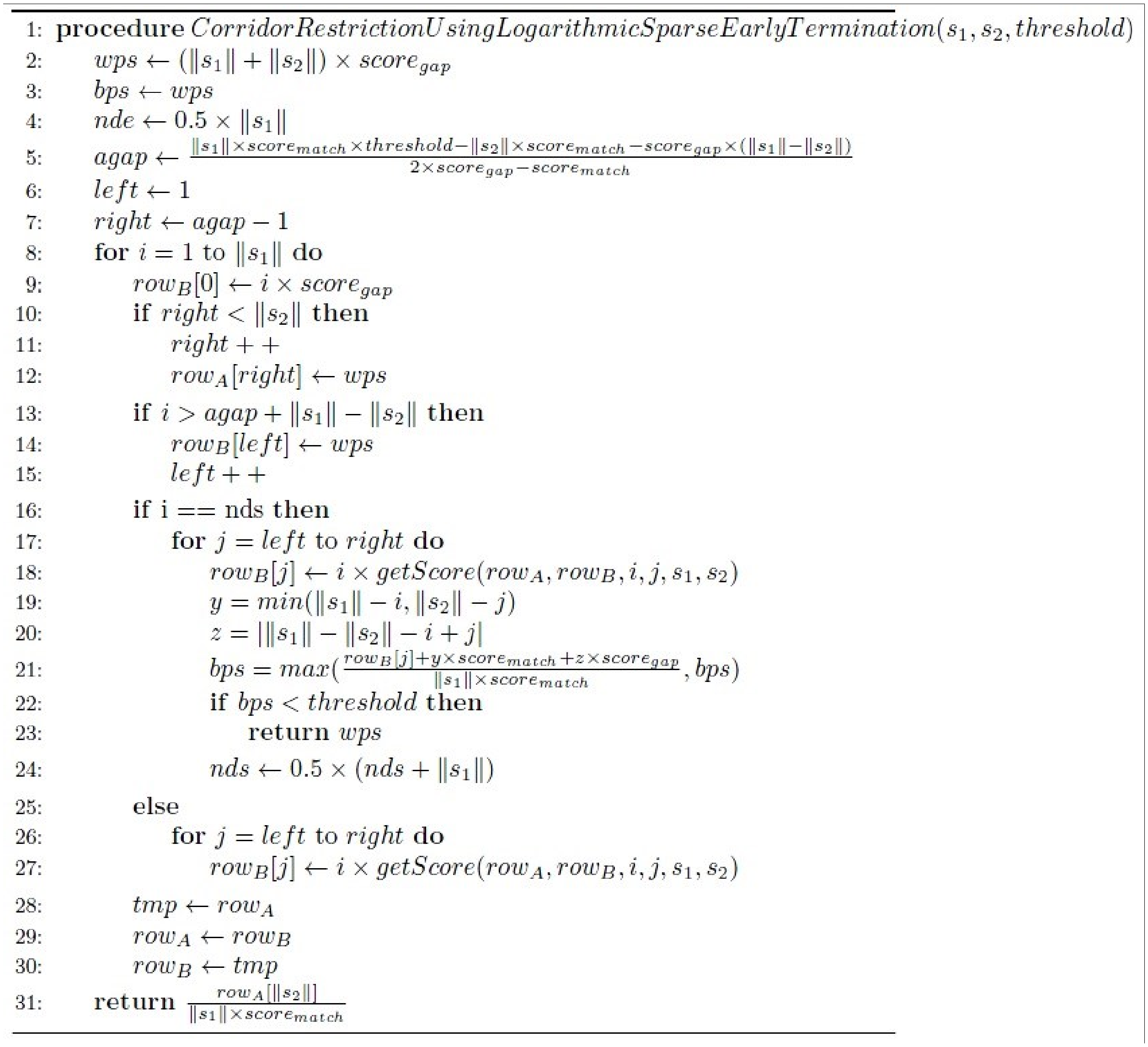


```
procedure $CorridorRestriction(s_1, s_2, threshold)$
    $wps \leftarrow (\|s_1\| + \|s_2\|) \times score_{gap}$
    $bps \leftarrow wps$
    $agap \leftarrow \frac{\|s_1\| \times score_{match} \times threshold - \|s_2\| \times score_{match} - score_{gap} \times (\|s_1\| - \|s_2\|)}{2 \times score_{gap} - score_{match}}$
    $left \leftarrow 1$
    $right \leftarrow agap - 1$
    for $i = 1$ to $\|s_1\|$ do
        $row_B[0] \leftarrow i \times score_{gap}$
        if $right < \|s_2\|$ then
            $right++$
            $row_A[right] \leftarrow wps$
        if $i > agap + \|s_1\| - \|s_2\|$ then
            $row_B[left] \leftarrow wps$
            $left++$
        for $j =$ left to right do
            $row_B[j] \leftarrow i \times getScore(row_A, row_B, i, j, s_1, s_2)$
        $tmp \leftarrow row_A$
        $row_A \leftarrow row_B$
        $row_B \leftarrow tmp$
    return $\frac{row_A[\|s_2\|]}{\|s_1\| \times score_{match}}$
```

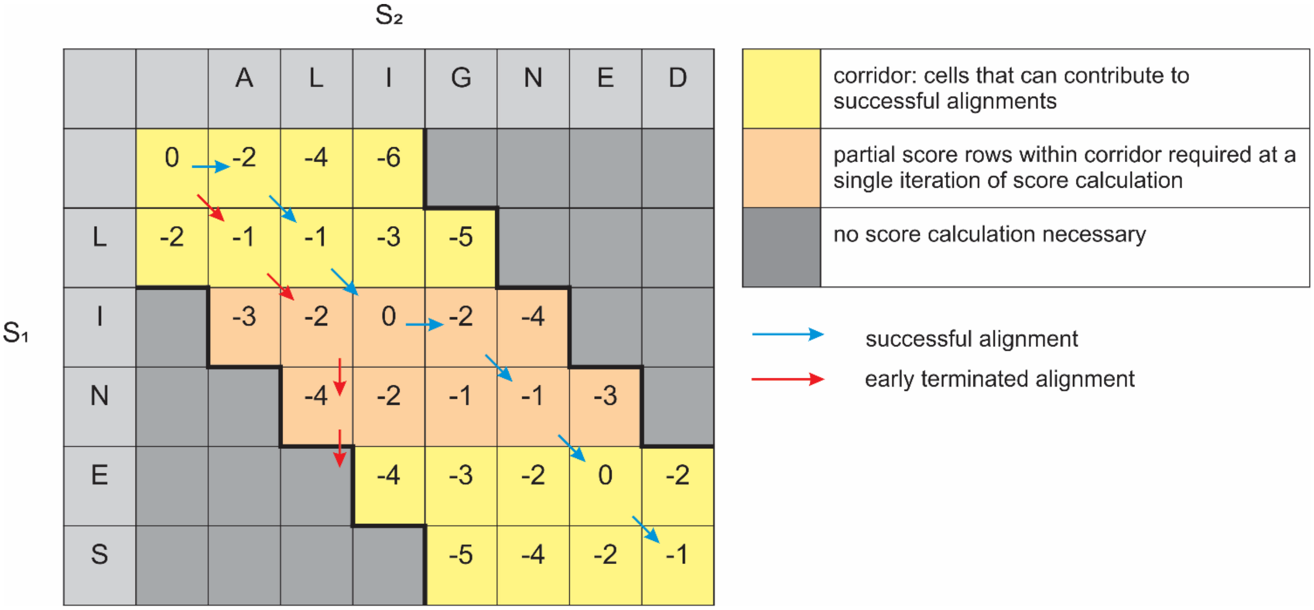
- $agap \Rightarrow$ allowed gap (deviation from center diagonal)
- $left \Rightarrow$ beginning position of corridor in $\|s_2\|$ for given position in $\|s_1\|$
- $right \Rightarrow$ ending position of corridor in $\|s_2\|$ for given position in $\|s_1\|$

Algorithm 2.6. can be further improved by using the corridor restriction along with logarithmic-sparse early-termination (Algorithm 2.7).

Algorithm 2.7. Corridor restriction with logarithmic-sparse early-termination algorithm:



```
procedure $CorridorRestrictionUsingLogarithmicSparseEarlyTermination(s_1, s_2, threshold)$
    $wps \leftarrow (\|s_1\| + \|s_2\|) \times score_{gap}$
    $bps \leftarrow wps$
    $nde \leftarrow 0.5 \times \|s_1\|$
    $agap \leftarrow \frac{\|s_1\| \times score_{match} \times threshold - \|s_2\| \times score_{match} - score_{gap} \times (\|s_1\| - \|s_2\|)}{2 \times score_{gap} - score_{match}}$
    $left \leftarrow 1$
    $right \leftarrow agap - 1$
    for $i = 1$ to $\|s_1\|$ do
        $row_B[0] \leftarrow i \times score_{gap}$
        if $right < \|s_2\|$ then
            $right++$
            $row_A[right] \leftarrow wps$
        if $i > agap + \|s_1\| - \|s_2\|$ then
            $row_B[left] \leftarrow wps$
            $left++$
        if i == nds then
            for $j = left$ to $right$ do
                $row_B[j] \leftarrow i \times getScore(row_A, row_B, i, j, s_1, s_2)$
                $y = min(\|s_1\| - i, \|s_2\| - j)$
                $z = |\|s_1\| - \|s_2\| - i + j|$
                $bps = max(\frac{row_B[j] + y \times score_{match} + z \times score_{gap}}{\|s_1\| \times score_{match}}, bps)$
                if $bps < threshold$ then
                    return $wps$
                $nds \leftarrow 0.5 \times (nds + \|s_1\|)$
        else
            for $j = left$ to $right$ do
                $row_B[j] \leftarrow i \times getScore(row_A, row_B, i, j, s_1, s_2)$
        $tmp \leftarrow row_A$
        $row_A \leftarrow row_B$
        $row_B \leftarrow tmp$
    return $\frac{row_A[\|s_2\|]}{\|s_1\| \times score_{match}}$
```



S₂
S₁
corridor: cells that can contribute to successful alignments
partial score rows within corridor required at a single iteration of score calculation
no score calculation necessary
successful alignment
early terminated alignment

		A	L	I	G	N	E	D
	0	-2	-4	-6				
L	-2	-1	-1	-3	-5			
I		-3	-2	0	-2	-4		
N			-4	-2	-1	-1	-3	
E				-4	-3	-2	0	-2
S					-5	-4	-2	-1

Figure 2.7. Corridor restriction and early-termination.

2.4.1 Performance test: Reduction in number of calculations by corridor restriction and early-termination approaches

To estimate the reduction of score calculations using corridor restriction and corridor restriction with early-termination, 19,900 pairwise alignments were performed using a subset of 200 randomly selected sequences of *Bartonella* spp. genes chosen with same sequence length range (between 500 and 600 nucleotides). The average percentage of calculations avoided using corridor restriction was found to be 83.10% ± 0.16%, when corridor restriction combined with logarithmic-sparse early-termination achieved 84.72% ± 1.87% reduction (Figure 2.8). In order to estimate the statistical significance of the improvement achieved in reducing the number of calculations required, the Wilcoxon matched-pairs signed rank test [33] was applied to the number of score calculations made by each algorithm for 19,900 pairwise alignments. Approximate p-value less than 0.0001 was obtained for each paired test. Each developed algorithm achieved a significant improvement over the original Hirschberg's algorithm.

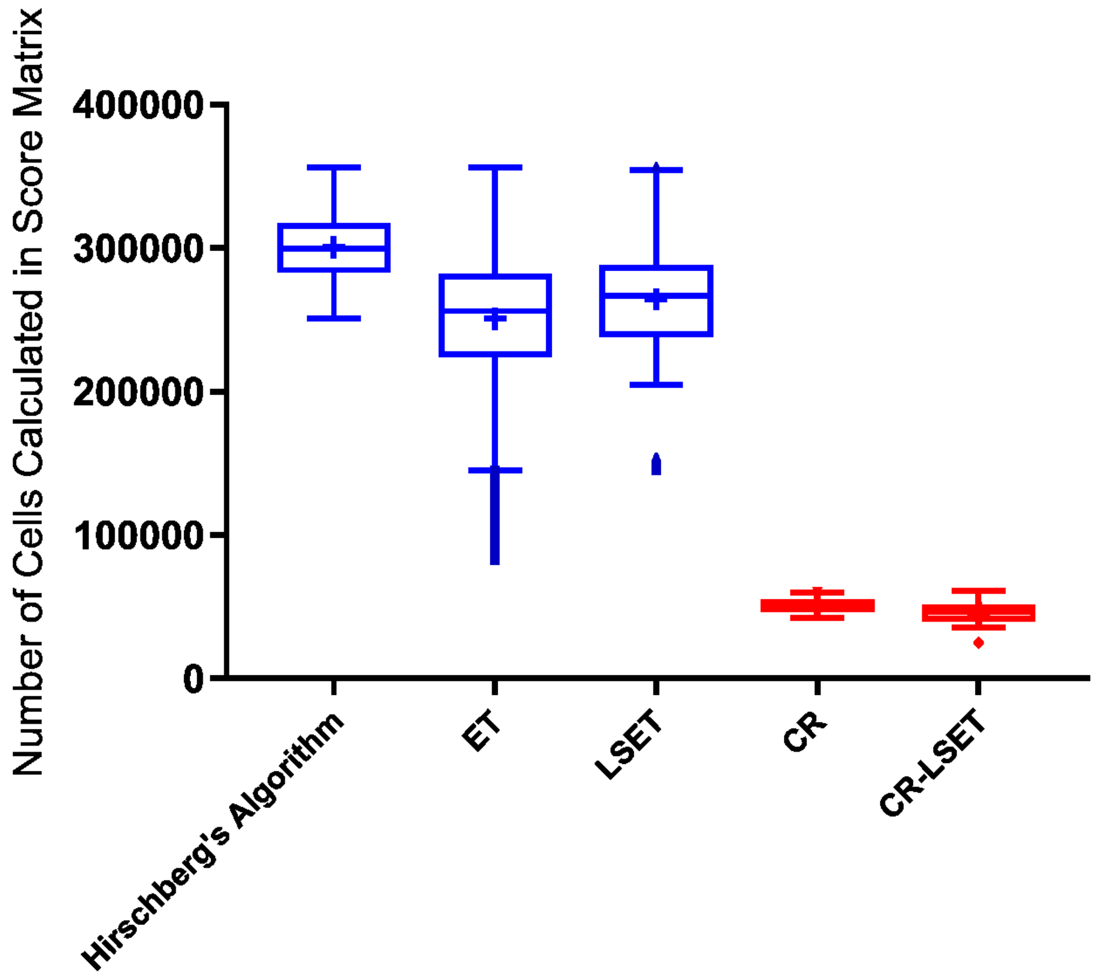


Figure 2.8. Performance comparison for early-termination (ET), logarithmic-sparse early-termination (LSET), corridor restriction (CR), and corridor restriction with logarithmic-sparse early-termination (CR-LSET) algorithms as compared to the Hirschberg's algorithm.

2.4.2 Performance test: Corridor restriction with early-termination running time

To estimate the real time gain using the corridor restriction with early-termination, 2,500 genes (~500 nucleotides) from *Bartonella* spp. were chosen for bootstrap testing. For each subset of gene sequences (Figure 2.9) the alignment experiment was performed five times. For a set of 400 genes, we observed ~6.5 times faster performance compared to the original Hirschberg's algorithm. With an additional number of alignments performed, the improvement can produce significant performance gains since the computational complexity of the algorithm was reduced to $\boldsymbol{O}(\boldsymbol{m}^{\mathbf{1}+\log_m agap})$**,** where m is sequence length and $agap$ is the allowed number of gaps between two sequences. The algorithm becomes more efficient for a smaller $agap$ (meaning that the required score threshold is higher), for the smallest possible value of $agap = 1$ (only pairwise alignment of identical sequences produce a sufficient score above a given threshold), time complexity becomes:

$$O\left(m^{1+\log_m 1}\right) \Rightarrow O(m^{1+0}) \Rightarrow O(m)$$

Since $agap \ll m$, then, $\boldsymbol{O}\left(\boldsymbol{m}^{\mathbf{1}+\log_m agap}\right) < \boldsymbol{O}(\boldsymbol{m}^{\mathbf{2}})$

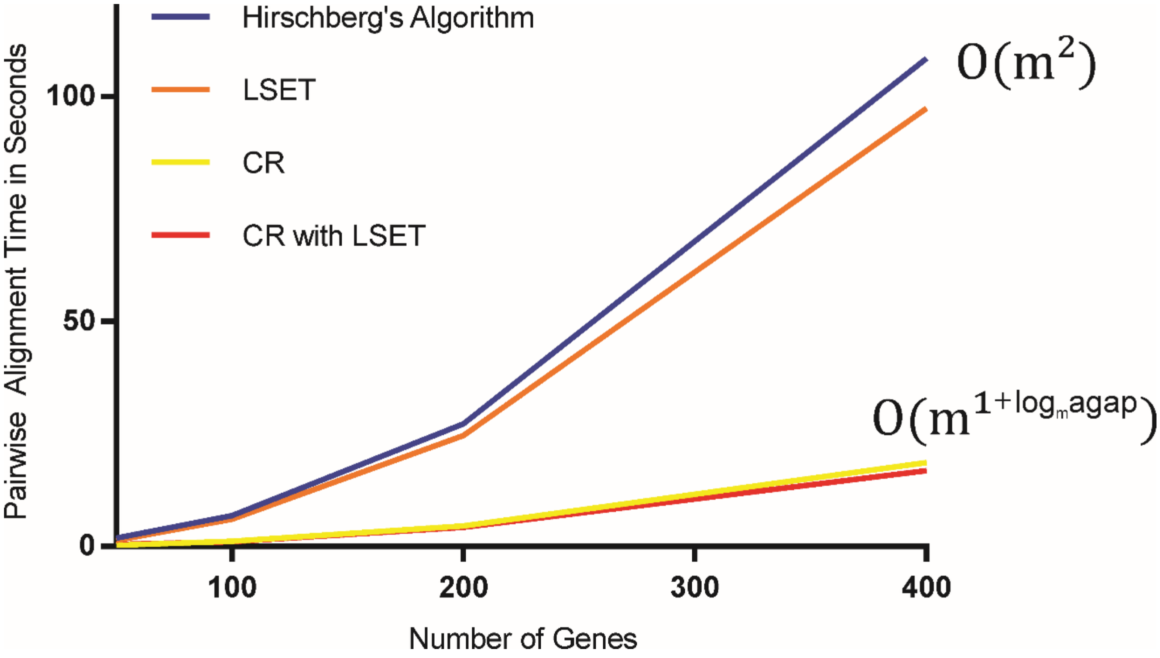


Figure 2.9. Run time performance comparison for the original Hirschberg's algorithm, logarithmic-sparse early-termination (LSET), corridor restriction (CR), and corridor restriction with logarithmic-sparse early-termination (CR with LSET) algorithms.

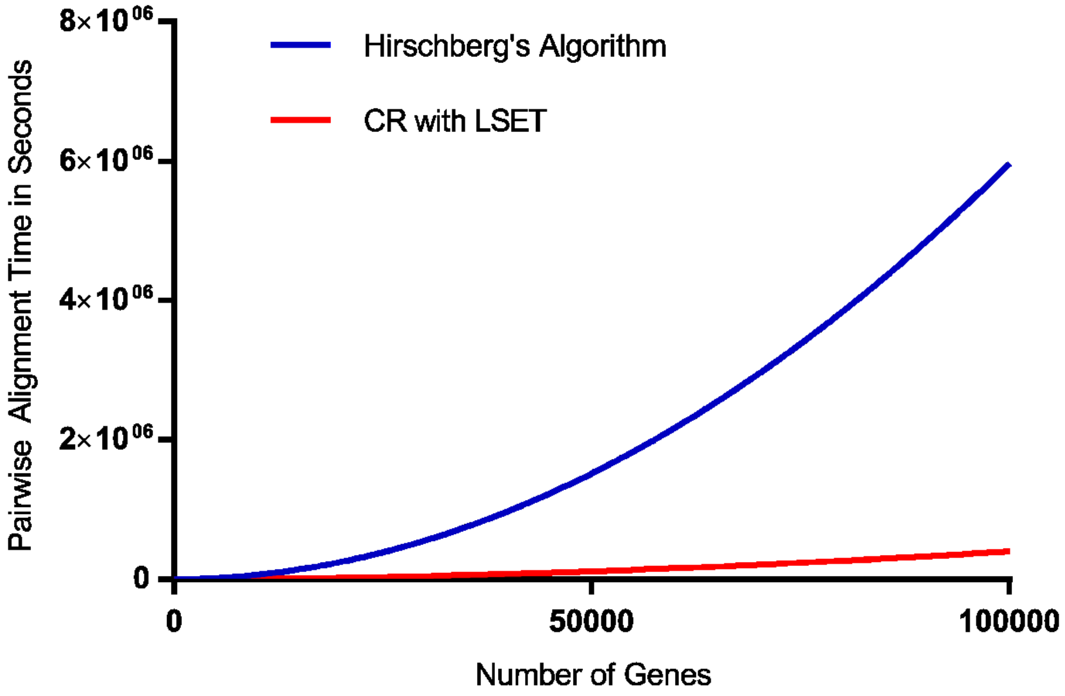


Figure 2.10. Run time performance comparison for original Hirschberg's algorithm[30]*, and corridor restriction with logarithmic-sparse early-termination (CR with LSET) algorithms extrapolated to pairwise alignment of 100,000 genes.*

Figure 2.10 illustrates the run time extrapolation of the proposed algorithm compared to Hirschberg's algorithm [30] for pairwise alignment of up to 100,000 genes (~5 billion pairwise alignments). The proposed algorithm is expected to complete ~5 billion pairwise alignment tasks in ~111 hours (~4.5 days), while the original Hirschberg's algorithm is expected to run more than ~1,659 hours (~69 days).

When all the approaches: diagonal-extension, logarithmic-sparse early-termination and corridor restriction were combined, the proposed algorithm improved the computational complexity of pairwise alignment from $O(m^2)$ to $O(m^{1+\log_m c})$. In a real case scenario, when hundreds of thousands of genes are clustered, improvement can make the difference between possible and impossible clustering tasks.

3. The Application of the Gene Clustering Algorithms to Characterize the Pan Genome of *Bartonella* spp.

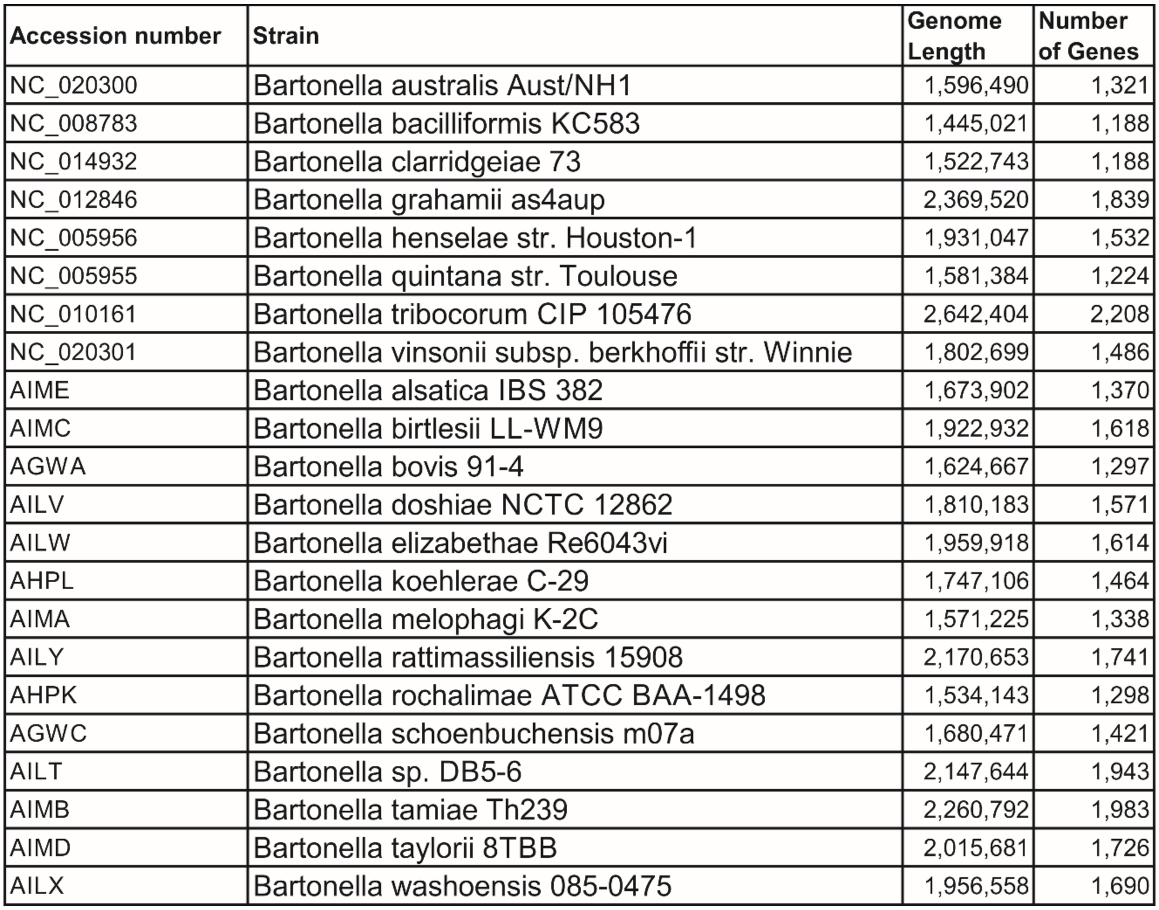
Bartonella is an important pathogen originating from South-East Asia and currently colonizing a variety of hosts on every continent, excluding Antarctica [34]. The usual mammal hosts include bats and rats [24]. It is believed that *Bartonella* spp. were first introduced to Africa and the Americas via rats traveling on ships [35]. Comparative analysis of *Bartonella* spp. genomes collected across different geographical locations from different hosts can reveal the evolution and any changes in the history of the spread of *Bartonella* across the globe.

Traditionally, the molecular clock approach [36] was used to estimate the evolutionary time separation among different strains of the same species: more mutations that accumulate in the same gene(s) from different two strains reflect a longer time of separation from their common ancestor. The mechanisms of bacterial adaptation however, are not limited by mutation accumulation but also includes genes lost and/or gained from lateral gene transfer. Regardless of the evolutionally separation time, the similarity in gene profiles among strains is expected to reflect functional similarity including adaptation to the host and its microbiome. Also adaptation to various external physical (temperature, humidity, UV light exposure) and biochemical conditions (pH, NaCl concentration) by mutation, permit the reconstruction of the history of its evolution/adaptation including bottlenecks, host changes, and acquisitions or loss of pathways. These changes reflect the surrounding microbiome history.

Clustering of the homologous genes across a number of known *Bartonella* genomes permit exploration of its evolution using functional similarities based upon the comparison of *Bartonella* gene profiles.

This study was performed using gene sequences from 22 complete or draft *Bartonella* spp. genomes available at NCBI [37] as of April 2015 (Table 3.1). The number of annotated genes in the genomes under consideration ranged between 1,188 and 2,208 (34,060 total). All the gene sequences were used in the presented analysis.

Table 3.1. List of 22 Bartonella genomes used in the analysis



Accession number	Strain	Genome Length	Number of Genes
NC_020300	Bartonella australis Aust/NH1	1,596,490	1,321
NC_008783	Bartonella bacilliformis KC583	1,445,021	1,188
NC_014932	Bartonella clarridgeiae 73	1,522,743	1,188
NC_012846	Bartonella grahamii as4aup	2,369,520	1,839
NC_005956	Bartonella henselae str. Houston-1	1,931,047	1,532
NC_005955	Bartonella quintana str. Toulouse	1,581,384	1,224
NC_010161	Bartonella tribocorum CIP 105476	2,642,404	2,208
NC_020301	Bartonella vinsonii subsp. berkhoffii str. Winnie	1,802,699	1,486
AIME	Bartonella alsatica IBS 382	1,673,902	1,370
AIMC	Bartonella birtlesii LL-WM9	1,922,932	1,618
AGWA	Bartonella bovis 91-4	1,624,667	1,297
AILV	Bartonella doshiae NCTC 12862	1,810,183	1,571
AILW	Bartonella elizabethae Re6043vi	1,959,918	1,614
AHPL	Bartonella koehlerae C-29	1,747,106	1,464
AIMA	Bartonella melophagi K-2C	1,571,225	1,338
AILY	Bartonella rattimassiliensis 15908	2,170,653	1,741
AHPK	Bartonella rochalimae ATCC BAA-1498	1,534,143	1,298
AGWC	Bartonella schoenbuchensis m07a	1,680,471	1,421
AILT	Bartonella sp. DB5-6	2,147,644	1,943
AIMB	Bartonella tamiae Th239	2,260,792	1,983
AIMD	Bartonella taylorii 8TBB	2,015,681	1,726
AILX	Bartonella washoensis 085-0475	1,956,558	1,690

3.1 Gene profile identification

The collection of *Bartonella* genomes used in the analysis contains strains significantly different from one another. Genome sizes vary almost two-fold: from 1.445 Mb (*Bartonella bacilliformis KC583*) to 2.642 Mb (*Bartonella tribocorum CIP 105476*). Not surprisingly, the same genes across different strains also appear dissimilar and the correct identification of same genes (gene families) had to be done carefully. To date, most attempts to identify “same” genes (homologous) in relatively diverse species have been based on establishing “reasonable criteria” -

the cutoff of the sequence-similarity score (in the nucleotide or amino acid space) below which genes are considered to be different [9], [14], [38]. Lack of a standard gene similarity threshold, various pan genome studies utilized different criteria to identify the homologous genes in relatively distant species. For example, in an *Escherichia coli* (E. *coli*) study by Rasko et al. [9], the amino acid identity threshold of approximately >80% was used. In a study by Lukjancenko et al. [14], two genes were attributed to a single gene family and considered 'conserved' when they shared at least a 50% amino acid identity over at least 50% of the length of the longest gene. Other studies consider a gene ‘conserved’ if the gene sequences shared over a 50% amino acid identity in conjunction with over 50% of the length of the longest gene [8]. Unfortunately, changing such criteria can significantly affect the outcome of the analysis.

3.2 *Bartonella* Genus specific scoring rules for pairwise alignment

The scoring rule which rewards perfect matches and penalizes mismatches defines the final score, but affects the number of gaps in the alignment. For example, equal penalties for gaps and substitutions lead to artificially higher numbers of gaps. The increased number of gaps do not reflect the nature of evolution of genomic sequences. It is also important to note that when single nucleotide substitutions are neutral (do not change the amino-acid sequence coded by this nucleotide) or in the worst case scenario, changes just one amino-acid in the resulting protein, gaps, usually resulting in frame shifting mutations, can dramatically alter the resulting protein product including artificial translation termination and/or cause non-sense mutations.

In order to identify *Bartonella* spp. specific scoring rules, we decided to estimate the average ratio between substitutions and gaps (InDels) by comparing randomly selected sequences of several known genes (Table 3.2, Figure 3.1). This analysis identified that the gap to substitution ratio is at least 1:10 and the “natural” scoring rules for *Bartonella* spp. are in range with previously published data [39]. For the purpose of this study, we chose the following scoring criteria: perfect match (+2), substitution (-1), and InDels (-10).

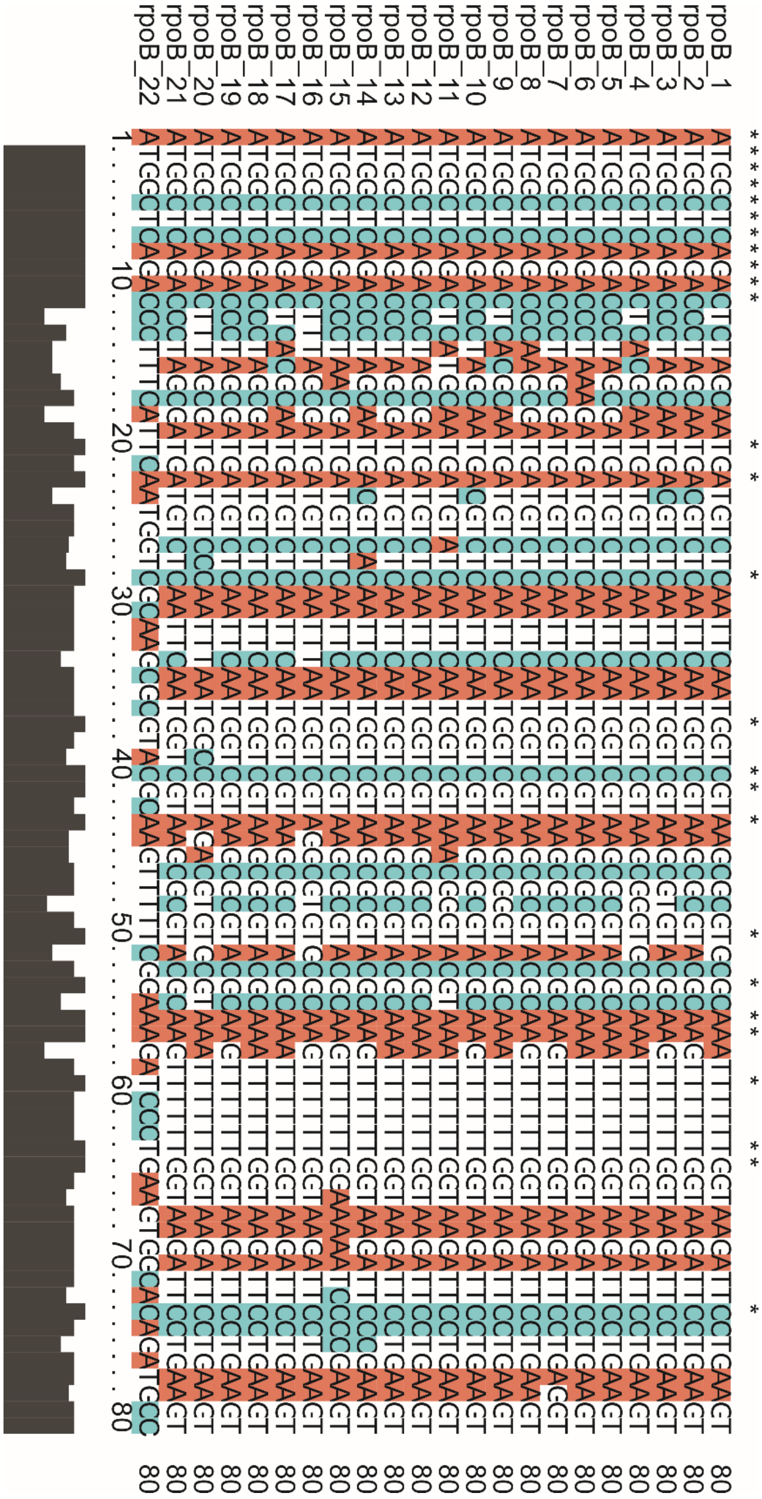
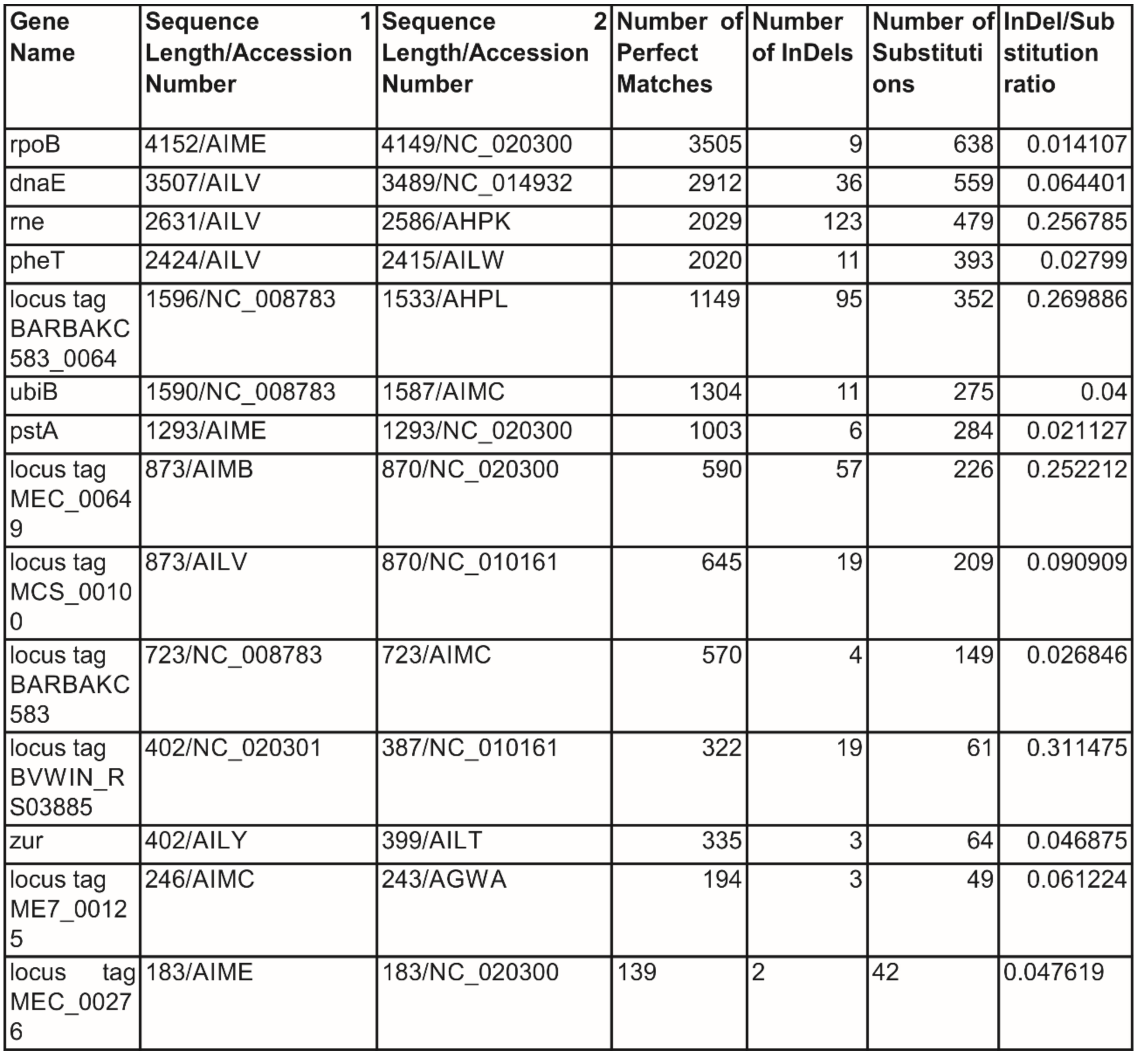


Figure 3.1 Multiple sequence alignment [40] *of the first 80 nucleotides of rpoB gene from 22 Bartonella spp. genomes.*

Table 3.2. InDels/Substitution ratio for selected pairs of Bartonella spp. genes. Average InDels/Substitution ratio is 0.1094



Gene Name	Sequence 1 Length/Accession Number	Sequence 2 Length/Accession Number	Number of Perfect Matches	Number of InDels	Number of Substitutions	InDel/Substitution ratio
rpoB	4152/AIME	4149/NC_020300	3505	9	638	0.014107
dnaE	3507/AILV	3489/NC_014932	2912	36	559	0.064401
rne	2631/AILV	2586/AHPK	2029	123	479	0.256785
pheT	2424/AILV	2415/AILW	2020	11	393	0.02799
locus tag BARBAKC583_0064	1596/NC_008783	1533/AHPL	1149	95	352	0.269886
ubiB	1590/NC_008783	1587/AIMC	1304	11	275	0.04
pstA	1293/AIME	1293/NC_020300	1003	6	284	0.021127
locus tag MEC_00649	873/AIMB	870/NC_020300	590	57	226	0.252212
locus tag MCS_00100	873/AILV	870/NC_010161	645	19	209	0.090909
locus tag BARBAKC583	723/NC_008783	723/AIMC	570	4	149	0.026846
locus tag BVWIN_RS03885	402/NC_020301	387/NC_010161	322	19	61	0.311475
zur	402/AILY	399/AILT	335	3	64	0.046875
locus tag ME7_00125	246/AIMC	243/AGWA	194	3	49	0.061224
locus tag MEC_00276	183/AIME	183/NC_020300	139	2	42	0.047619

In order to introduce a “natural” sequence-similarity cutoff in the nucleotide space, we performed an alignment for each pair of gene sequences of all 22 *Bartonella* spp. under consideration (a total of 580,024,770 pairs) using algorithms presented in chapters 1 and 2.

As expected, the distribution of the alignment scores for each pair of genes (Figure 3.2) appeared to have two local maxima. The distribution on the left with a local maximum alignment score value of around 0.02 represents all the dissimilar genes. It is important to note that due to the negative penalties for InDels and substitutions, alignments can have negative values. The distribution on the right, which is more “diverse”, represents the set of sequences (genes) with a significant level of similarity. It is also necessary to emphasize that the left hand side distribution is also located in the area of alignment scores for random sequences (Figure 3.2). As one can see, the distribution of alignment scores also has a well pronounced divider between two maximums, grouping “different” and “similar” pairs of genes located between alignment score values of 0.1 and 0.3. The cutoff value must to be taken from this interval.

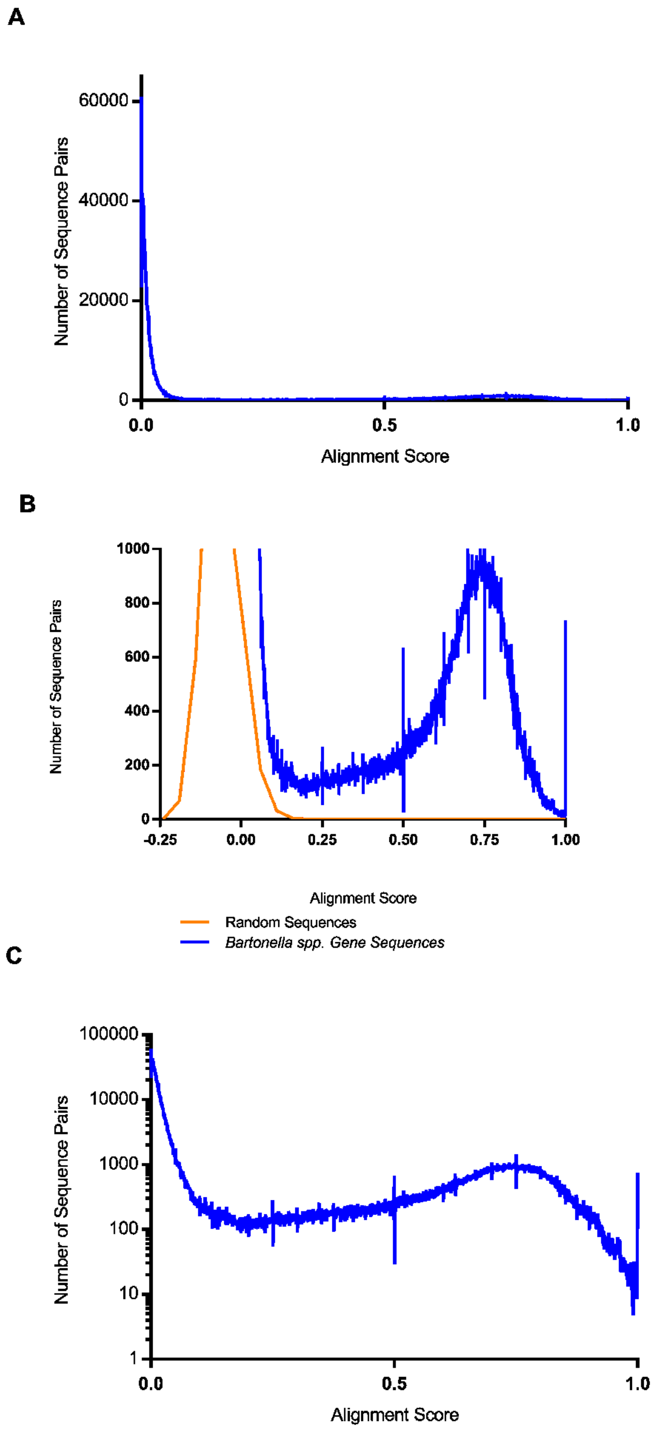


Figure 3.2. Alignment score distribution for each pair of 34,060 genes from 22 Bartonella genomes as well as random sequences of various lengths: (A) full distribution; (B) zoom on Y-axis; (C) same distribution in log scale.

To more accurately identify the appropriate cutoff; we decided to evaluate how the cutoff value would affect the number of genes conserved across the *Bartonella* spp. (core genome). The idea behind this analysis is that the total number of genes identified as the core genome is a result of two competing processes. An increasing cutoff value reduces the number of common genes by assigning even slightly different sequences to be different. On the other hand, decreasing the threshold will cluster dissimilar genes, so in extreme cases they all become a single cluster.

Figure 3.3 and Table 3.3 show the dependence of the number of core genes common for all 22 *Bartonella* genomes for a chosen threshold. A similar analysis of 11 *E. coli* reference genomes is seen on Table 3.4. Based on the analysis, we chose the threshold to be 0.15. As seen in Figure 3.3 the shape of the core-genome-size function for *E. coli* is different. For higher cutoff values, a large flat region reflects *E. coli* genomes are more similar to each other than the *Bartonella* genomes.

Table 3.3. Dependence of the number of unique genes conserved across 22 Bartonella spp. genomes to the alignment-score threshold

Alignment Score Threshold	0.12	0.13	0.14	0.15	0.16	0.17	0.18	0.19	0.2	0.21	0.22	0.23	0.24	0.25
Core Genome Size	712	714	714	714	712	707	703	696	695	688	687	687	682	680

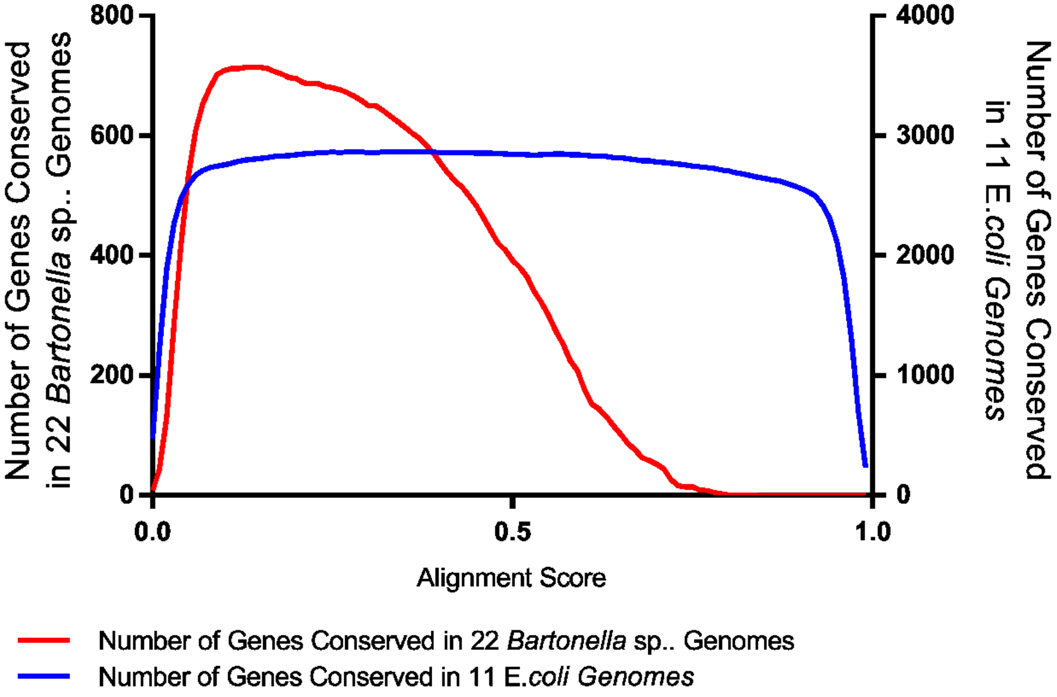
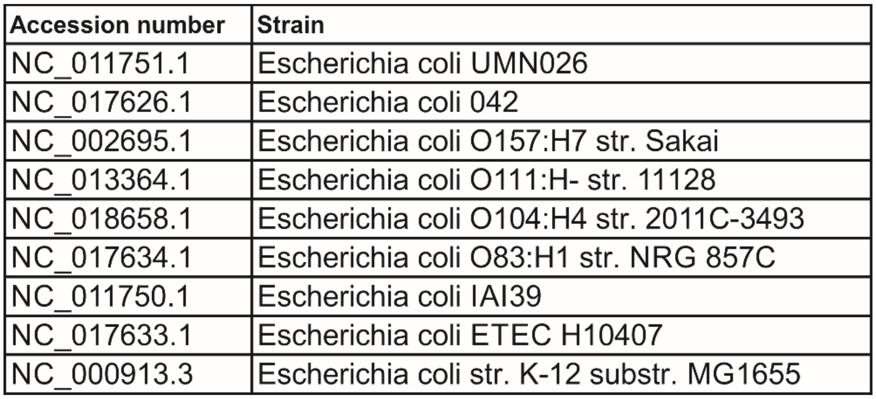


Figure 3.3. Core-genome size for 22 Bartonella and 11 E. coli genomes as a function of the alignment-score cutoff.

Table 3.4. E.coli genomes used in the analysis



Accession number	**Strain**
NC_011751.1	Escherichia coli UMN026
NC_017626.1	Escherichia coli 042
NC_002695.1	Escherichia coli O157:H7 str. Sakai
NC_013364.1	Escherichia coli O111:H- str. 11128
NC_018658.1	Escherichia coli O104:H4 str. 2011C-3493
NC_017634.1	Escherichia coli O83:H1 str. NRG 857C
NC_011750.1	Escherichia coli IAI39
NC_017633.1	Escherichia coli ETEC H10407
NC_000913.3	Escherichia coli str. K-12 substr. MG1655

It is also important to mention that in some cases, when a vast majority of the genes were clustered correctly, the chosen threshold clustered several relatively similar genes together. Examples seen in Figures 3.4a and 3.4b show several examples.

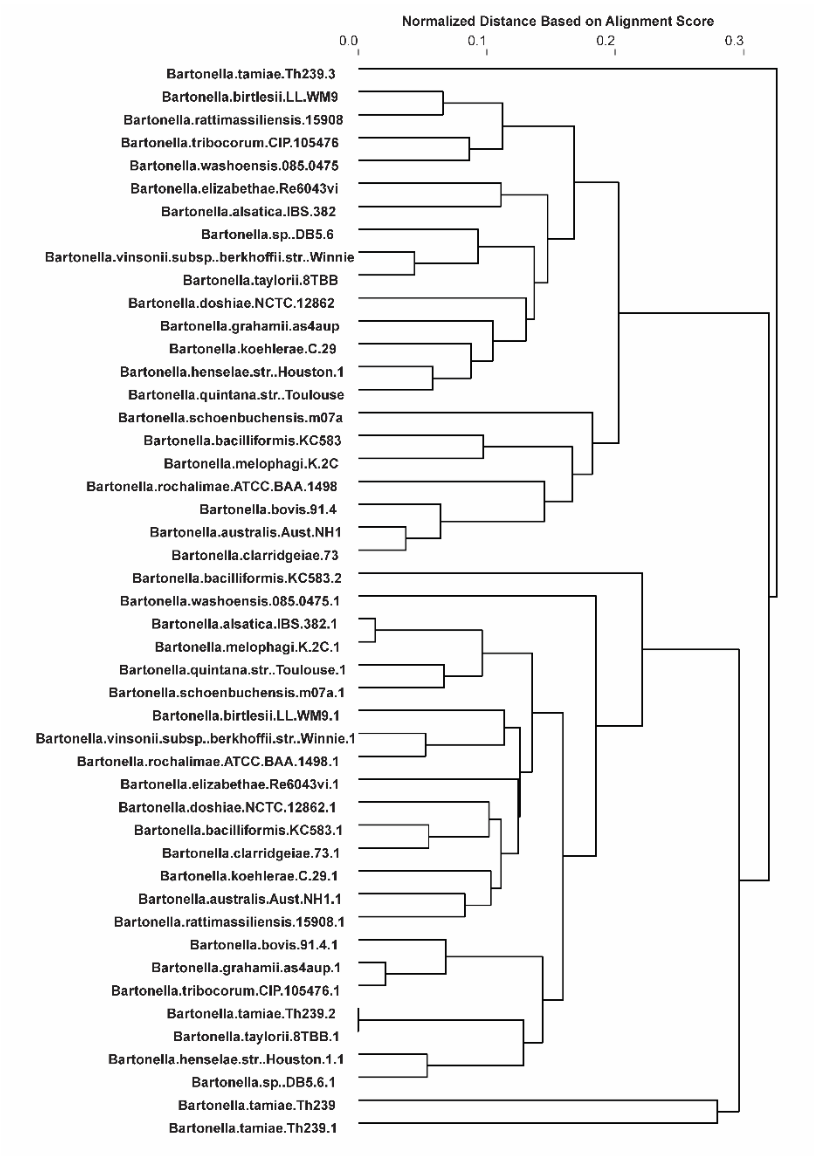

Normalized Distance Based on Alignment Score
0.0
0.1
0.2
0.3
Bartonella.tamiae.Th239.3
Bartonella.birtlesii.LL.WM9
Bartonella.rattimassiliensis.15908
Bartonella.tribocorum.CIP.105476
Bartonella.washoensis.085.0475
Bartonella.elizabethae.Re6043vi
Bartonella.alsatica.IBS.382
Bartonella.sp..DB5.6
Bartonella.vinsonii.subsp..berkhoffii.str..Winnie
Bartonella.taylorii.8TBB
Bartonella.doshiae.NCTC.12862
Bartonella.grahamii.as4aup
Bartonella.koehlerae.C.29
Bartonella.henselae.str..Houston.1
Bartonella.quintana.str..Toulouse
Bartonella.schoenbuchensis.m07a
Bartonella.bacilliformis.KC583
Bartonella.melophagi.K.2C
Bartonella.rochalimae.ATCC.BAA.1498
Bartonella.bovis.91.4
Bartonella.australis.Aust.NH1
Bartonella.clarridgeiae.73
Bartonella.bacilliformis.KC583.2
Bartonella.washoensis.085.0475.1
Bartonella.alsatica.IBS.382.1
Bartonella.melophagi.K.2C.1
Bartonella.quintana.str..Toulouse.1
Bartonella.schoenbuchensis.m07a.1
Bartonella.birtlesii.LL.WM9.1
Bartonella.vinsonii.subsp..berkhoffii.str..Winnie.1
Bartonella.rochalimae.ATCC.BAA.1498.1
Bartonella.elizabethae.Re6043vi.1
Bartonella.doshiae.NCTC.12862.1
Bartonella.bacilliformis.KC583.1
Bartonella.clarridgeiae.73.1
Bartonella.koehlerae.C.29.1
Bartonella.australis.Aust.NH1.1
Bartonella.rattimassiliensis.15908.1
Bartonella.bovis.91.4.1
Bartonella.grahamii.as4aup.1
Bartonella.tribocorum.CIP.105476.1
Bartonella.tamiae.Th239.2
Bartonella.taylorii.8TBB.1
Bartonella.henselae.str..Houston.1.1
Bartonella.sp..DB5.6.1
Bartonella.tamiae.Th239
Bartonella.tamiae.Th239.1

Figure 3.4a Phylogenetic tree based on nucleotide similarity among gltL (glutamate-aspartate ABC transporter ATP-binding component) and glnQ (amino acid ABC transporter, ATP-binding protein) sequences placed in the same cluster using a threshold of 0.15.

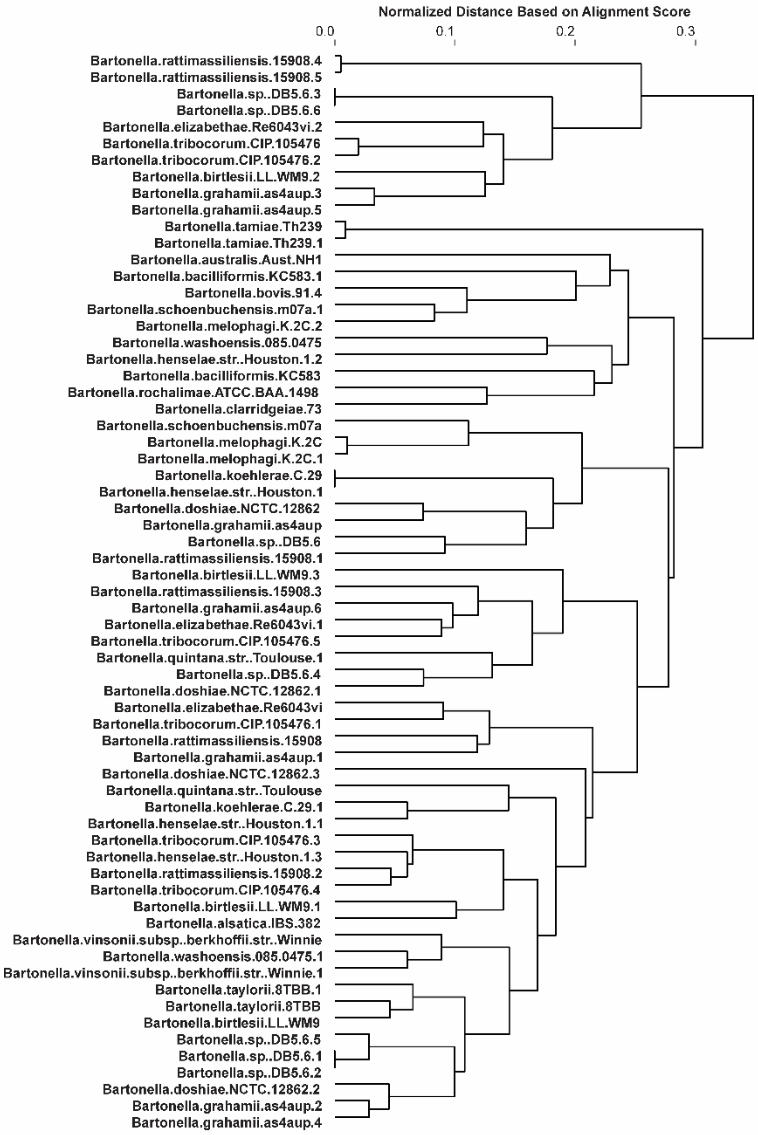

Normalized Distance Based on Alignment Score
0.0
0.1
0.2
0.3
Bartonella.rattimassiliensis.15908.4
Bartonella.rattimassiliensis.15908.5
Bartonella.sp..DB5.6.3
Bartonella.sp..DB5.6.6
Bartonella.elizabethae.Re6043vi.2
Bartonella.tribocorum.CIP.105476
Bartonella.tribocorum.CIP.105476.2
Bartonella.birtlesii.LL.WM9.2
Bartonella.grahamii.as4aup.3
Bartonella.grahamii.as4aup.5
Bartonella.tamiae.Th239
Bartonella.tamiae.Th239.1
Bartonella.australis.Aust.NH1
Bartonella.bacilliformis.KC583.1
Bartonella.bovis.91.4
Bartonella.schoenbuchensis.m07a.1
Bartonella.melophagi.K.2C.2
Bartonella.washoensis.085.0475
Bartonella.henselae.str..Houston.1.2
Bartonella.bacilliformis.KC583
Bartonella.rochalimae.ATCC.BAA.1498
Bartonella.clarridgeiae.73
Bartonella.schoenbuchensis.m07a
Bartonella.melophagi.K.2C
Bartonella.melophagi.K.2C.1
Bartonella.koehlerae.C.29
Bartonella.henselae.str..Houston.1
Bartonella.doshiae.NCTC.12862
Bartonella.grahamii.as4aup
Bartonella.sp..DB5.6
Bartonella.rattimassiliensis.15908.1
Bartonella.birtlesii.LL.WM9.3
Bartonella.rattimassiliensis.15908.3
Bartonella.grahamii.as4aup.6
Bartonella.elizabethae.Re6043vi.1
Bartonella.tribocorum.CIP.105476.5
Bartonella.quintana.str..Toulouse.1
Bartonella.sp..DB5.6.4
Bartonella.doshiae.NCTC.12862.1
Bartonella.elizabethae.Re6043vi
Bartonella.tribocorum.CIP.105476.1
Bartonella.rattimassiliensis.15908
Bartonella.grahamii.as4aup.1
Bartonella.doshiae.NCTC.12862.3
Bartonella.quintana.str..Toulouse
Bartonella.koehlerae.C.29.1
Bartonella.henselae.str..Houston.1.1
Bartonella.tribocorum.CIP.105476.3
Bartonella.henselae.str..Houston.1.3
Bartonella.rattimassiliensis.15908.2
Bartonella.tribocorum.CIP.105476.4
Bartonella.birtlesii.LL.WM9.1
Bartonella.alsatica.IBS.382
Bartonella.vinsonii.subsp..berkhoffii.str..Winnie
Bartonella.washoensis.085.0475.1
Bartonella.vinsonii.subsp..berkhoffii.str..Winnie.1
Bartonella.taylorii.8TBB.1
Bartonella.taylorii.8TBB
Bartonella.birtlesii.LL.WM9
Bartonella.sp..DB5.6.5
Bartonella.sp..DB5.6.1
Bartonella.sp..DB5.6.2
Bartonella.doshiae.NCTC.12862.2
Bartonella.grahamii.as4aup.2
Bartonella.grahamii.as4aup.4

Figure 3.4b Phylogenetic tree based on nucleotide similarity among sequences inside of the cluster containing phage related lysozyme sequences.

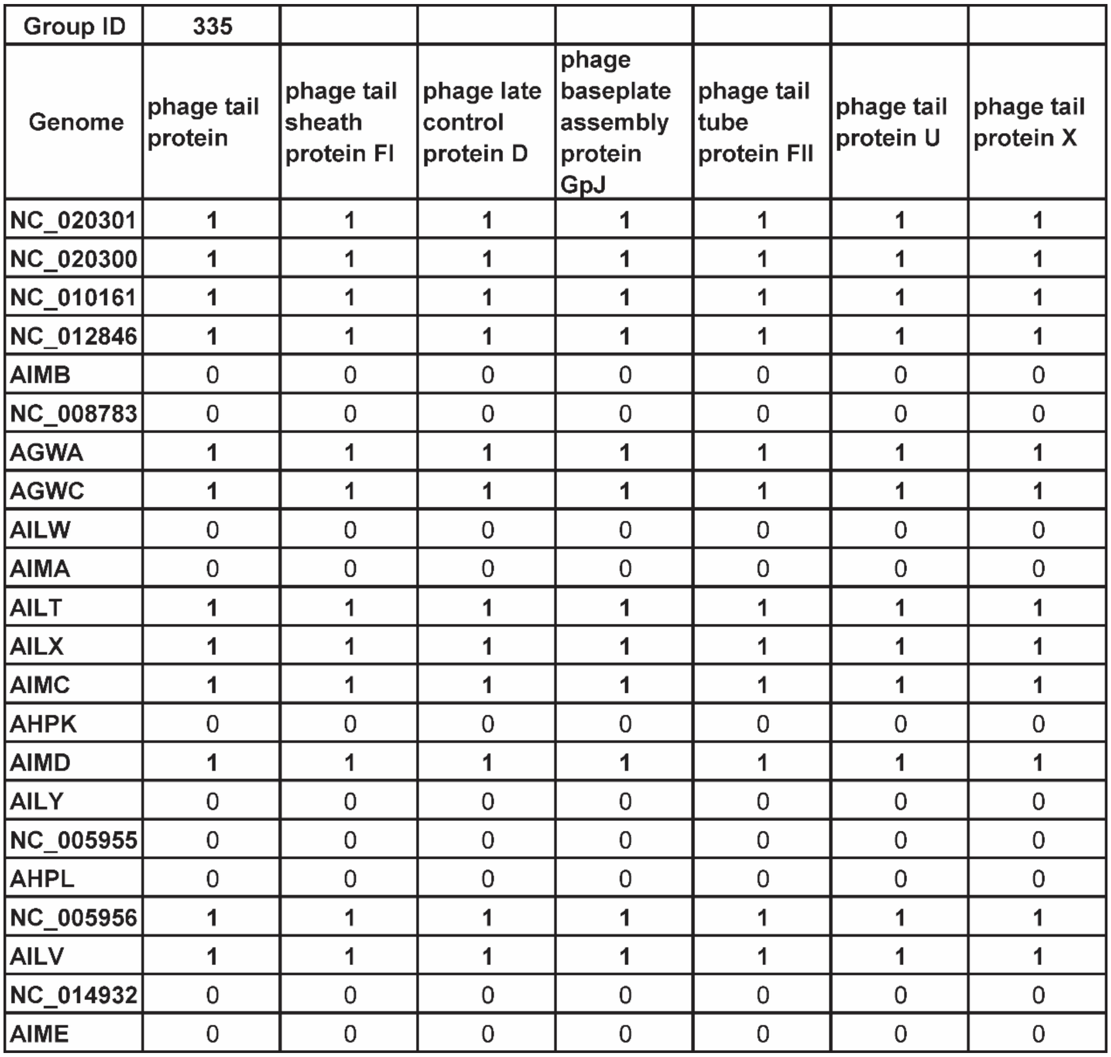
3.3 Results

The analysis performed using the chosen clustering parameters provided the identification of 6,043 gene clusters out of 34,060 total gene sequences. It also resulted in the presence/absence and copy-number profile for every gene and gene cluster in each *Bartonella* genome used in the analysis. The subset of the pan genome across all 22 *Bartonella* spp. contains a total of 6,043 different genes with 714 gene clusters conserved across all genomes. These are potential genes in the core genome. Interestingly, some of the core-genome genes were found to be present in more than one copy (up to 18 copies) in *Bartonella* genomes.

Analysis of the gene-cluster profiles of *Bartonella* genomes provide an identification of correlated gene-cluster patterns. Table 3.5 shows examples of such groups. Such co-appearances of genes could indicate similar metabolic or signaling pathways (Table 3.5). Figure 3.4 shows the location of six different groups (335, 349, 399, 400, 541, and 544) on the gene-similarity based tree (also listed in Tables 3.5 and 3.6).

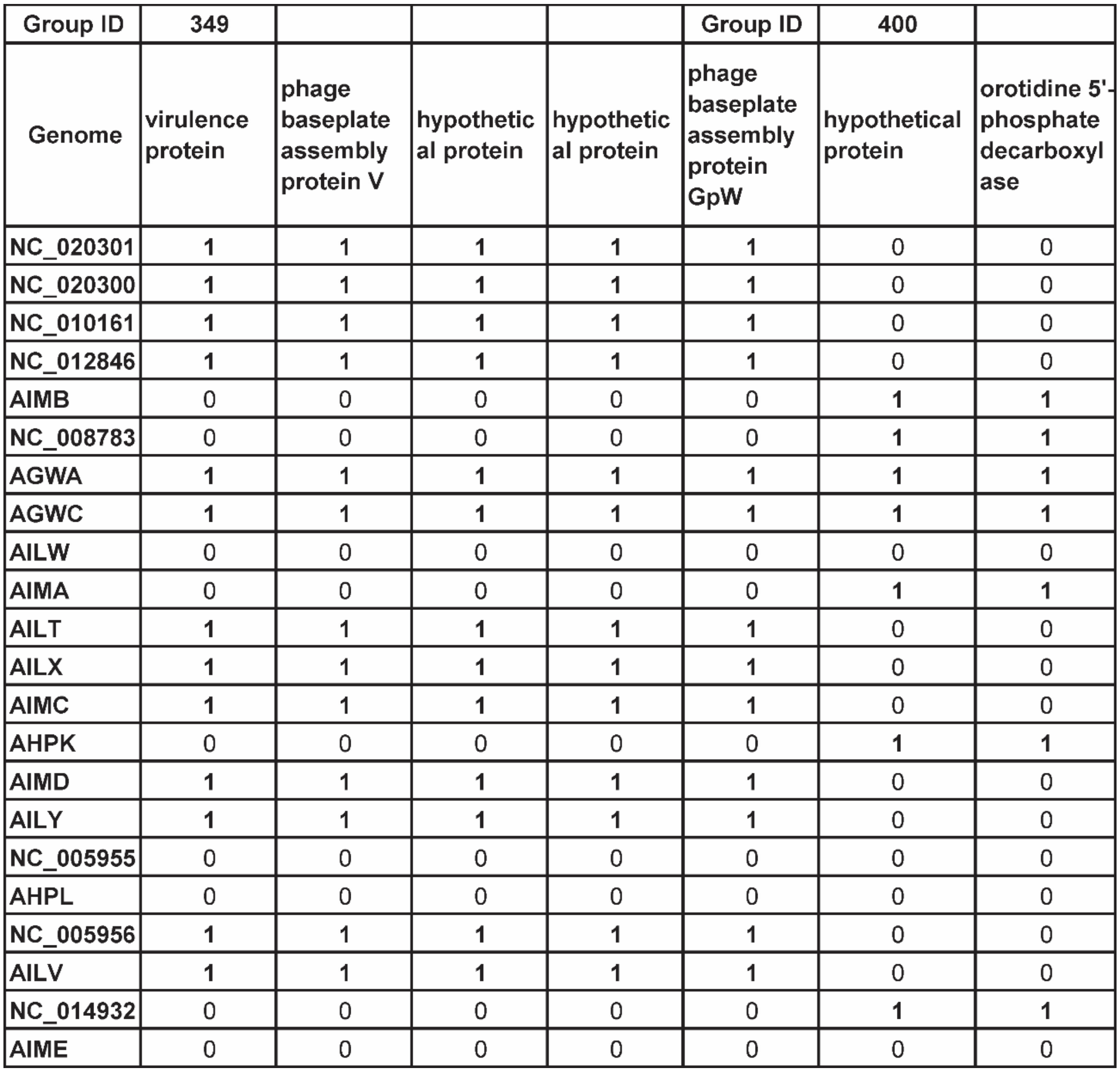
Table 3.5 Co-appearance pattern of genes for groups 335 (a), 349 and 400 (b) and 541 (c)

a.



Group ID	335						
Genome	phage tail protein	phage tail sheath protein FI	phage late control protein D	phage baseplate assembly protein GpJ	phage tail tube protein FII	phage tail protein U	phage tail protein X
NC_020301	1	1	1	1	1	1	1
NC_020300	1	1	1	1	1	1	1
NC_010161	1	1	1	1	1	1	1
NC_012846	1	1	1	1	1	1	1
AIMB	0	0	0	0	0	0	0
NC_008783	0	0	0	0	0	0	0
AGWA	1	1	1	1	1	1	1
AGWC	1	1	1	1	1	1	1
AILW	0	0	0	0	0	0	0
AIMA	0	0	0	0	0	0	0
AILT	1	1	1	1	1	1	1
AILX	1	1	1	1	1	1	1
AIMC	1	1	1	1	1	1	1
AHPK	0	0	0	0	0	0	0
AIMD	1	1	1	1	1	1	1
AILY	0	0	0	0	0	0	0
NC_005955	0	0	0	0	0	0	0
AHPL	0	0	0	0	0	0	0
NC_005956	1	1	1	1	1	1	1
AILV	1	1	1	1	1	1	1
NC_014932	0	0	0	0	0	0	0
AIME	0	0	0	0	0	0	0

b.



Group ID	349				Group ID	400	
Genome	virulence protein	phage baseplate assembly protein V	hypothetical protein	hypothetical protein	phage baseplate assembly protein GpW	hypothetical protein	orotidine 5'-phosphate decarboxylase
NC_020301	1	1	1	1	1	0	0
NC_020300	1	1	1	1	1	0	0
NC_010161	1	1	1	1	1	0	0
NC_012846	1	1	1	1	1	0	0
AIMB	0	0	0	0	0	1	1
NC_008783	0	0	0	0	0	1	1
AGWA	1	1	1	1	1	1	1
AGWC	1	1	1	1	1	1	1
AILW	0	0	0	0	0	0	0
AIMA	0	0	0	0	0	1	1
AILT	1	1	1	1	1	0	0
AILX	1	1	1	1	1	0	0
AIMC	1	1	1	1	1	0	0
AHPK	0	0	0	0	0	1	1
AIMD	1	1	1	1	1	0	0
AILY	1	1	1	1	1	0	0
NC_005955	0	0	0	0	0	0	0
AHPL	0	0	0	0	0	0	0
NC_005956	1	1	1	1	1	0	0
AILV	1	1	1	1	1	0	0
NC_014932	0	0	0	0	0	1	1
AIME	0	0	0	0	0	0	0

c.

Group ID	541								
Genome	type VI secretion protein	trwE protein	TrwD protein	orotidine 5'-phosphate decarboxylase	P-type conjugative transfer protein VirB9	hypothetical protein	hypothetical protein	KorA protein	KorB protein
NC_020301	1	1	1	1	1	1	1	1	1
NC_020300	1	1	1	1	1	1	1	1	1
NC_010161	1	1	1	1	1	1	1	1	1
NC_012846	1	1	1	1	1	1	1	1	1
AIMB	0	0	0	0	0	0	0	0	0
NC_008783	0	0	0	0	0	0	0	0	0
AGWA	0	0	0	0	0	0	0	0	0
AGWC	0	0	0	0	0	0	0	0	0
AILW	1	1	1	1	1	1	1	1	1
AIMA	0	0	0	0	0	0	0	0	0
AILT	1	1	1	1	1	1	1	1	1
AILX	1	1	1	1	1	1	1	1	1
AIMC	1	1	1	1	1	1	1	1	1
AHPK	0	0	0	0	0	0	0	0	0
AIMD	1	1	1	1	1	1	1	1	1
AILY	1	1	1	1	1	1	1	1	1
NC_005955	1	1	1	1	1	1	1	1	1
AHPL	1	1	1	1	1	1	1	1	1
NC_005956	1	1	1	1	1	1	1	1	1
AILV	1	1	1	1	1	1	1	1	1
NC_014932	0	0	0	0	0	0	0	0	0
AIME	1	1	1	1	1	1	1	1	1

Some groups appearing in genomes also appeared to be anti-correlated, that is, genes belonging to two different groups never appeared in the same genome. This may indicate that the processes controlled by these gene clusters cannot be performed together (Table 3.6).

Table 3.6. Example of an avoidance pattern demonstrating that a group of genes involved in flagella development (group 399) and heat-shock (group 544) proteins pathways may not coincide in Bartonella spp.

Group ID	399		Group ID	544			
Genome	flagellar hook protein FlgK	flagellar hook protein FlgL	Flagellar transcriptional regulator ftcR	hypothetical protein	NAD(P)H-dependent glycerol-3-phosphate dehydrogenase	peptidoglycan-binding protein LysM	heat-shock protein Hsp20
NC_020301	0	0	0	1	1	1	1
NC_020300	0	0	0	1	1	1	1
NC_010161	0	0	0	1	1	1	1
NC_012846	0	0	0	1	1	1	1
AIMB	0	0	0	1	1	1	1
NC_008783	1	1	1	0	0	0	0
AGWA	1	1	1	0	0	0	0
AGWC	1	1	1	0	0	0	0
AILW	0	0	0	1	1	1	1
AIMA	1	1	1	0	0	0	0
AILT	0	0	0	1	1	1	1
AILX	0	0	0	1	1	1	1
AIMC	0	0	0	1	1	1	1
AHPK	1	1	1	0	0	0	0
AIMD	0	0	0	1	1	1	1
AILY	0	0	0	1	1	1	1
NC_005955	0	0	0	1	1	1	1
AHPL	0	0	0	1	1	1	1
NC_005956	0	0	0	1	1	1	1
AILV	0	0	0	1	1	1	1
NC_014932	1	1	1	0	0	0	0
AIME	0	0	0	1	1	1	1

Using the Jaccard index [41] (number of unique common genes over the total number of unique genes present in at least one of two genomes), a gene profile similarity tree of *Bartonella* spp. was created using a complete linkage-tree algorithm (Figure 3.5). This figure shows that *Bartonella* spp. are organized into two general groups and *Bartonella tamiae Th239* is distant from the other

Bartonella spp. based on its gene profile. Every group and organism has several unique genes. Values associated in the tree represent the number of gene clusters unique/common/total for each branch. The core-genome size (number of common genes) of the first group starting with *Bartonella melophagi K2C* and ending with *Bartonella alsatica IBS 382* is 803 while a total number of 3,000 genes were identified for this group. This group is organized into two subdivisions, with the left most subdivision containing a group of genes (group 399) that only appears in genomes belonging to this subdivision. Interestingly, this gene group is responsible for the development of flagella. This subdivision is very distinct from the rest of the *Bartonella* spp. in the sense that it does not contain genes from group 544 while all other genomes contain this group. Group 544 contains LysM [42], [43] (protein shown to bind to other bacterial cell walls and chitin – found in insect exoskeletons) and Hsp20 [44] (a heat-induced stress protein).

The other division of genomes starting with *Bartonella rattimassilliensis 15908* and ending with *Bartonella taylorii 8TBB* have 1,041 genes in common where the total number of genes (3,607) is larger than the first group. This may be due to the number of genomes in each group, the first group contains 12 genomes while the latter contains 9, with *Bartonella tamiae* an outlier.

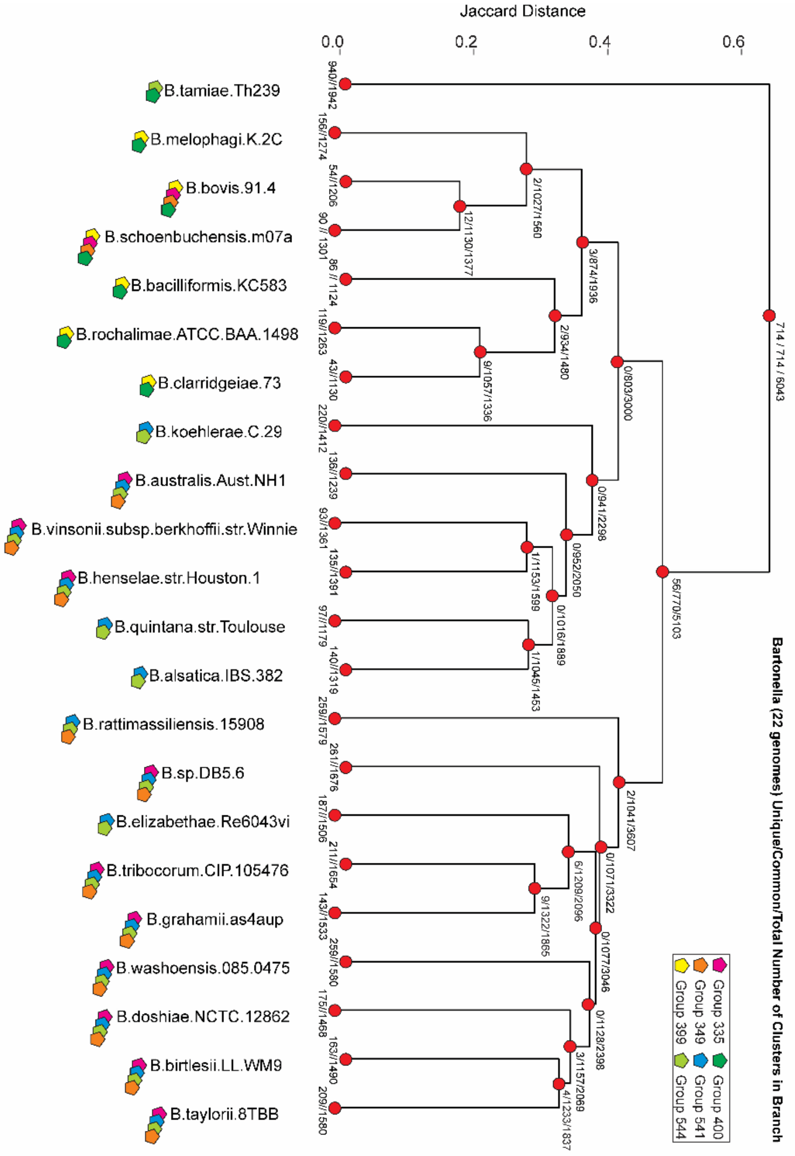


*Figure 3.5. Gene profile similarity tree for 22 Bartonella genomes. Values associated to each of the tree branches include number of genes **Unique** (present in all genomes inside the branch but absent in any other genome) / **Common** (present in all genomes inside the branch and may also be present in other genomes) / **Total** (present in at least one genome inside the branch).*

The conserved genes used in analysis are 16S rRNA (ribosomal RNA), rpoC (DNA-directed RNA polymerase subunit beta), aatA (aspartate aminotransferase), dapD (2,3,4,5-tetrahydropyridine-2,6-carboxylate N-succinyltransferase), rpsB (30S ribosomal protein S2), and tmk (thymidylate kinase). As expected, the topology of the tree appears to be different from the trees generated using sequence-similarity from few conserved genes (Figure 3.6a-g). For example, *Bartonella koehlerae C29* and *Bartonella Henselae Houston 1* genomes appear to be similar based on the trees using the genes: rpoC (Figure 3.6b), aatA (Figure 3.6c), dapD (Figure 3.6d), rpsB (Figure 3.6e), tmk (Figure 3.6f). The consensus tree based on these genes (Figure 3.6g) and the consensus tree based on all 714 core genes (Figure 3.7) show a similar result. In the gene-profile-based tree (Figure 3.5), however, *Bartonella koehlerae C29* appears to be distant from all other genomes in its sub-division, and *Bartonella Henselae Houston 1* appears to be closest to *Bartonella vinsonii subsp. Berkhoffii Winnie.* In the trees based on these genes *Bartonella vinsonii subsp. Berkhoffii Winnie* appears to be dissimilar to both previously mentioned genomes.

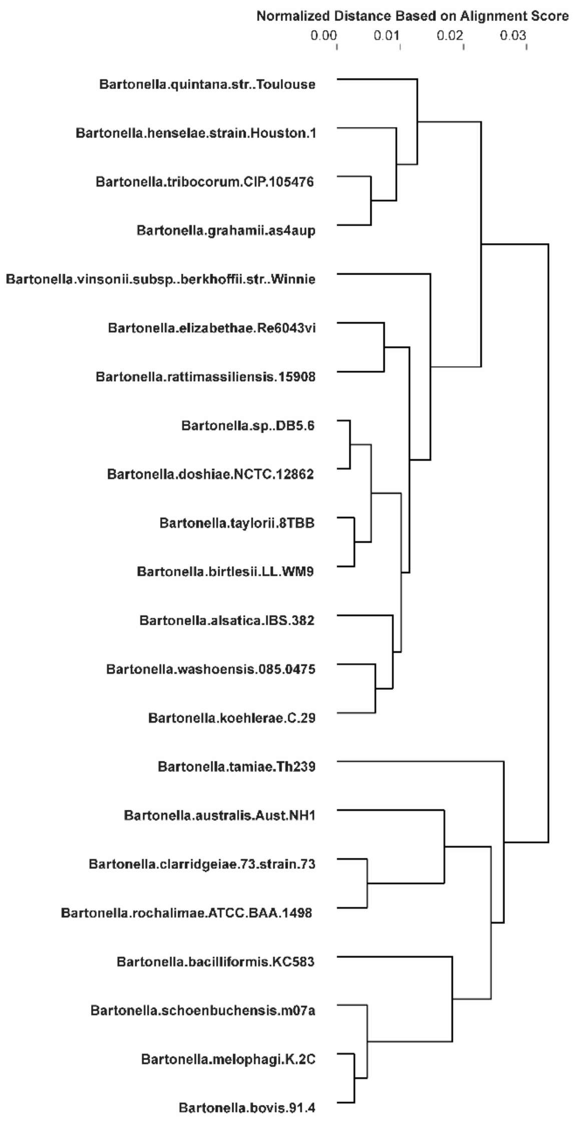

Normalized Distance Based on Alignment Score
0.00
0.01
0.02
0.03
Bartonella.quintana.str..Toulouse
Bartonella.henselae.strain.Houston.1
Bartonella.tribocorum.CIP.105476
Bartonella.grahamii.as4aup
Bartonella.vinsonii.subsp..berkhoffii.str..Winnie
Bartonella.elizabethae.Re6043vi
Bartonella.rattimassiliensis.15908
Bartonella.sp..DB5.6
Bartonella.doshiae.NCTC.12862
Bartonella.taylorii.8TBB
Bartonella.birtlesii.LL.WM9
Bartonella.alsatica.IBS.382
Bartonella.washoensis.085.0475
Bartonella.koehlerae.C.29
Bartonella.tamiae.Th239
Bartonella.australis.Aust.NH1
Bartonella.clarridgeiae.73.strain.73
Bartonella.rochalimae.ATCC.BAA.1498
Bartonella.bacilliformis.KC583
Bartonella.schoenbuchensis.m07a
Bartonella.melophagi.K.2C
Bartonella.bovis.91.4

Figure 3.6a. Phylogenetic tree based on nucleotide similarity among single copies of 16S rRNA (ribosomal RNA) present in 22 Bartonella spp.

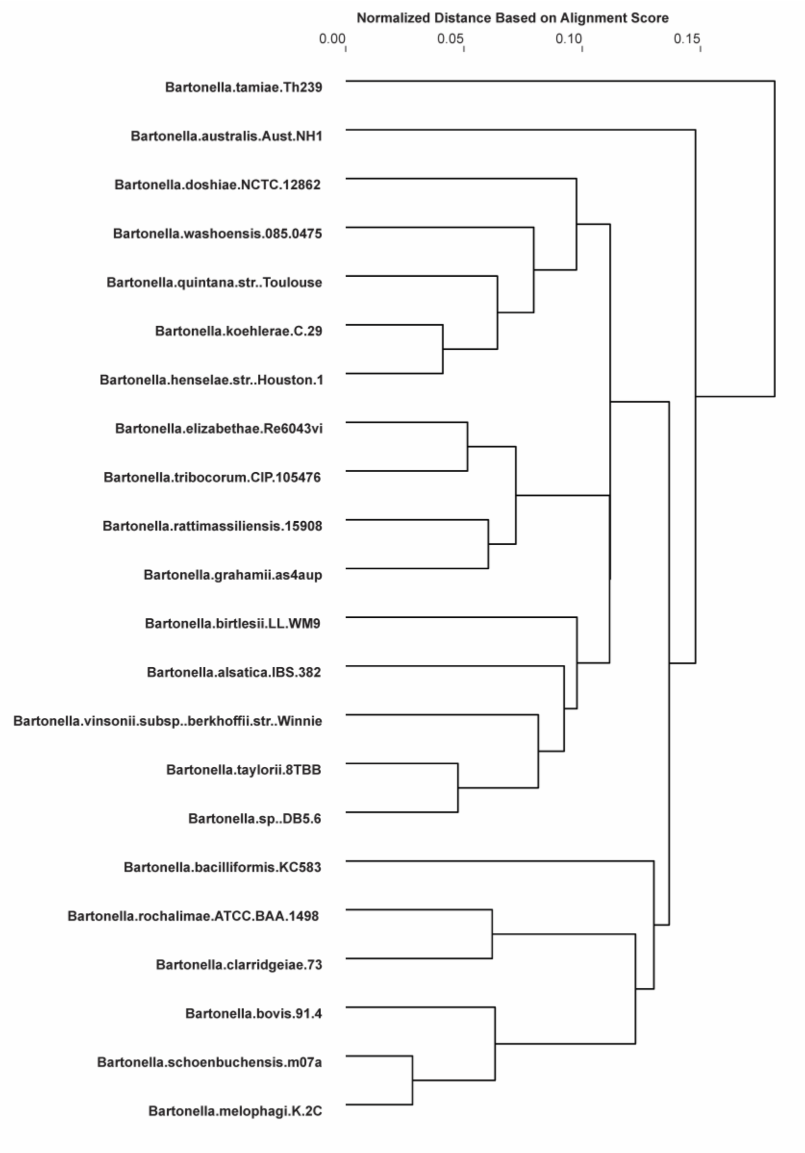

Normalized Distance Based on Alignment Score
0.00
0.05
0.10
0.15
Bartonella.tamiae.Th239
Bartonella.australis.Aust.NH1
Bartonella.doshiae.NCTC.12862
Bartonella.washoensis.085.0475
Bartonella.quintana.str..Toulouse
Bartonella.koehlerae.C.29
Bartonella.henselae.str..Houston.1
Bartonella.elizabethae.Re6043vi
Bartonella.tribocorum.CIP.105476
Bartonella.rattimassiliensis.15908
Bartonella.grahamii.as4aup
Bartonella.birtlesii.LL.WM9
Bartonella.alsatica.IBS.382
Bartonella.vinsonii.subsp..berkhoffii.str..Winnie
Bartonella.taylorii.8TBB
Bartonella.sp..DB5.6
Bartonella.bacilliformis.KC583
Bartonella.rochalimae.ATCC.BAA.1498
Bartonella.clarridgeiae.73
Bartonella.bovis.91.4
Bartonella.schoenbuchensis.m07a
Bartonella.melophagi.K.2C

Figure 3.6b. Phylogenetic tree based on nucleotide similarity among rpoC (DNA-directed RNA polymerase subunit beta) genes.

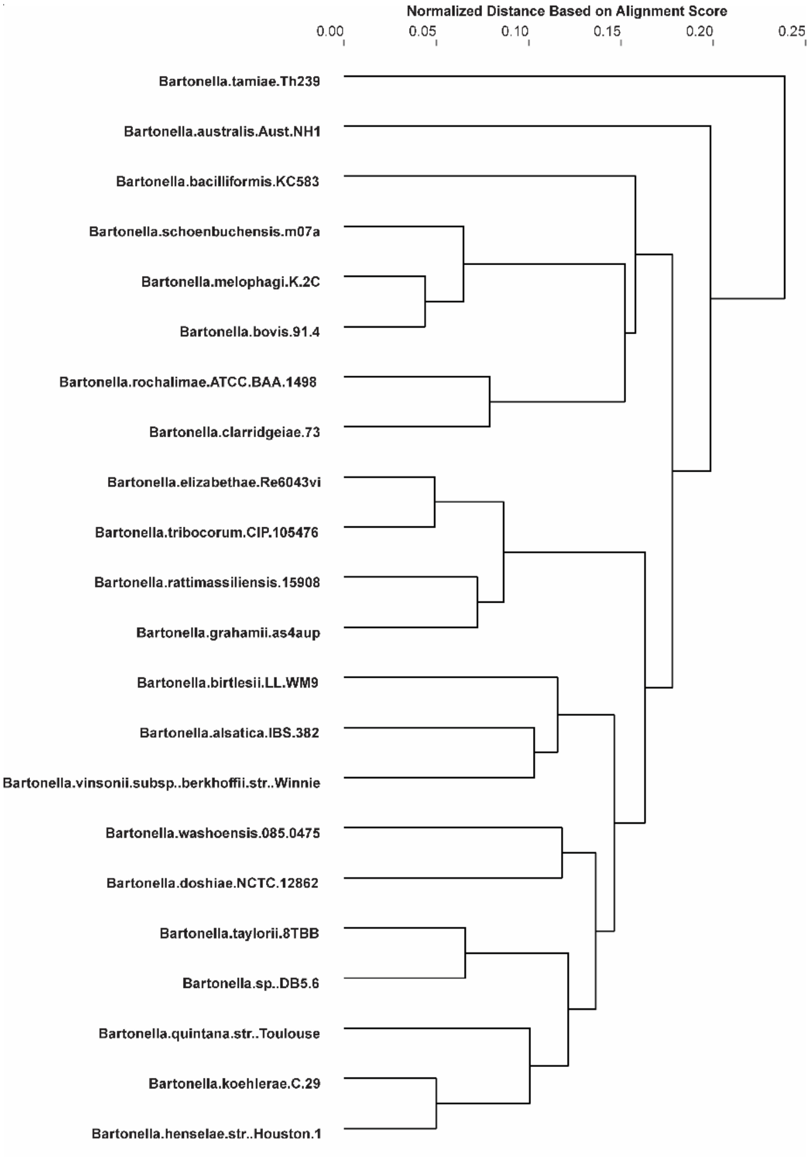

Normalized Distance Based on Alignment Score
0.00
0.05
0.10
0.15
0.20
0.25
Bartonella.tamiae.Th239
Bartonella.australis.Aust.NH1
Bartonella.bacilliformis.KC583
Bartonella.schoenbuchensis.m07a
Bartonella.melophagi.K.2C
Bartonella.bovis.91.4
Bartonella.rochalimae.ATCC.BAA.1498
Bartonella.clarridgeiae.73
Bartonella.elizabethae.Re6043vi
Bartonella.tribocorum.CIP.105476
Bartonella.rattimassiliensis.15908
Bartonella.grahamii.as4aup
Bartonella.birtlesii.LL.WM9
Bartonella.alsatica.IBS.382
Bartonella.vinsonii.subsp..berkhoffii.str..Winnie
Bartonella.washoensis.085.0475
Bartonella.doshiae.NCTC.12862
Bartonella.taylorii.8TBB
Bartonella.sp..DB5.6
Bartonella.quintana.str..Toulouse
Bartonella.koehlerae.C.29
Bartonella.henselae.str..Houston.1

Figure 3.6c. Phylogenetic tree based on nucleotide similarity among aatA (aspartate aminotransferase) genes.

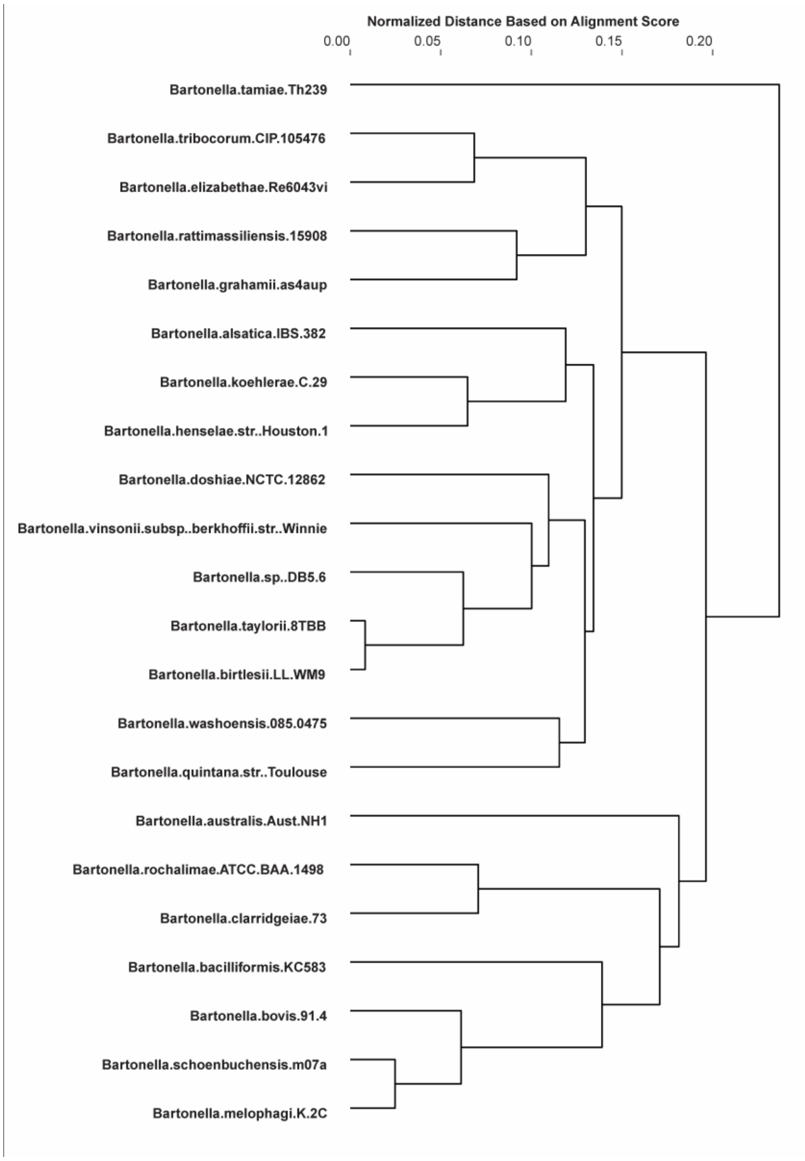

Normalized Distance Based on Alignment Score
0.00
0.05
0.10
0.15
0.20
Bartonella.tamiae.Th239
Bartonella.tribocorum.CIP.105476
Bartonella.elizabethae.Re6043vi
Bartonella.rattimassiliensis.15908
Bartonella.grahamii.as4aup
Bartonella.alsatica.IBS.382
Bartonella.koehlerae.C.29
Bartonella.henselae.str..Houston.1
Bartonella.doshiae.NCTC.12862
Bartonella.vinsonii.subsp..berkhoffii.str..Winnie
Bartonella.sp..DB5.6
Bartonella.taylorii.8TBB
Bartonella.birtlesii.LL.WM9
Bartonella.washoensis.085.0475
Bartonella.quintana.str..Toulouse
Bartonella.australis.Aust.NH1
Bartonella.rochalimae.ATCC.BAA.1498
Bartonella.clarridgeiae.73
Bartonella.bacilliformis.KC583
Bartonella.bovis.91.4
Bartonella.schoenbuchensis.m07a
Bartonella.melophagi.K.2C

Figure 3.6d. Phylogenetic tree based on nucleotide similarity among dapD (2,3,4,5-tetrahydropyridine-2,6-carboxylate N-succinyltransferase) genes.

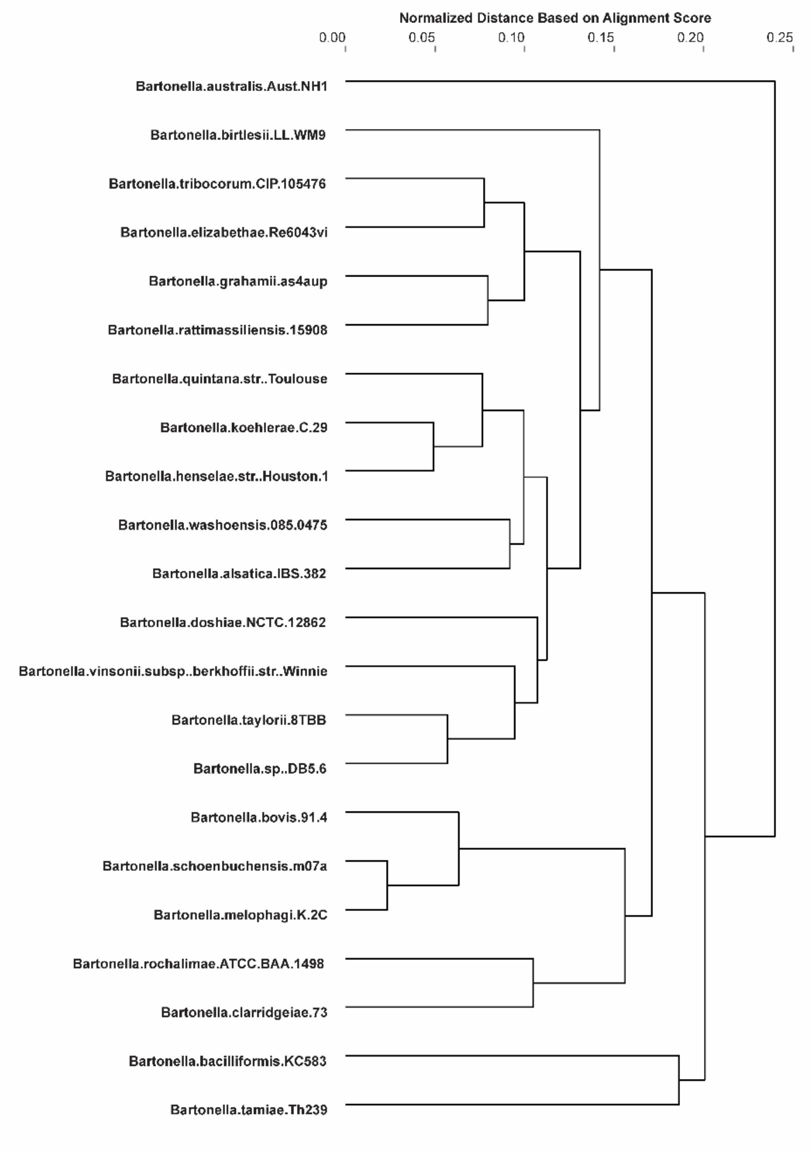

Normalized Distance Based on Alignment Score
0.00
0.05
0.10
0.15
0.20
0.25
Bartonella.australis.Aust.NH1
Bartonella.birtlesii.LL.WM9
Bartonella.tribocorum.CIP.105476
Bartonella.elizabethae.Re6043vi
Bartonella.grahamii.as4aup
Bartonella.rattimassiliensis.15908
Bartonella.quintana.str..Toulouse
Bartonella.koehlerae.C.29
Bartonella.henselae.str..Houston.1
Bartonella.washoensis.085.0475
Bartonella.alsatica.IBS.382
Bartonella.doshiae.NCTC.12862
Bartonella.vinsonii.subsp..berkhoffii.str..Winnie
Bartonella.taylorii.8TBB
Bartonella.sp..DB5.6
Bartonella.bovis.91.4
Bartonella.schoenbuchensis.m07a
Bartonella.melophagi.K.2C
Bartonella.rochalimae.ATCC.BAA.1498
Bartonella.clarridgeiae.73
Bartonella.bacilliformis.KC583
Bartonella.tamiae.Th239

Figure 3.6e. Phylogenetic tree based on nucleotide similarity among rpsB (30S ribosomal protein S2) genes.

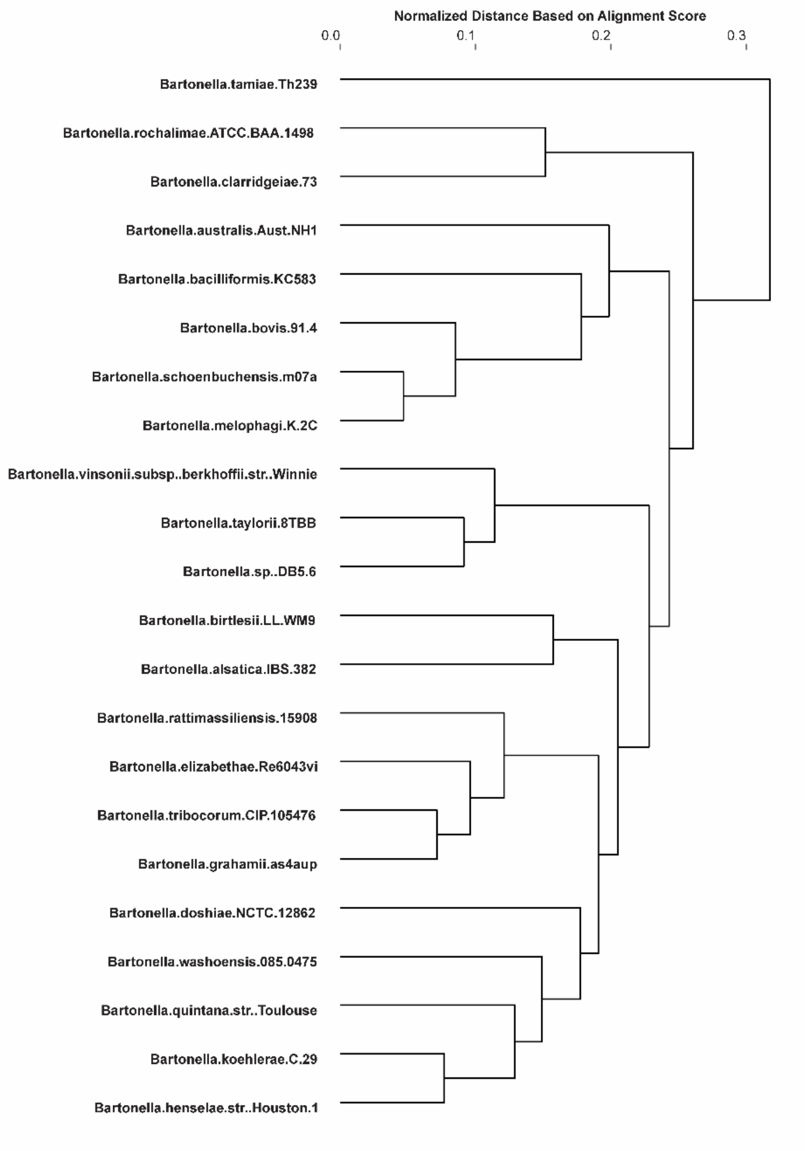

Normalized Distance Based on Alignment Score
0.0
0.1
0.2
0.3
Bartonella.tamiae.Th239
Bartonella.rochalimae.ATCC.BAA.1498
Bartonella.clarridgeiae.73
Bartonella.australis.Aust.NH1
Bartonella.bacilliformis.KC583
Bartonella.bovis.91.4
Bartonella.schoenbuchensis.m07a
Bartonella.melophagi.K.2C
Bartonella.vinsonii.subsp..berkhoffii.str..Winnie
Bartonella.taylorii.8TBB
Bartonella.sp..DB5.6
Bartonella.birtlesii.LL.WM9
Bartonella.alsatica.IBS.382
Bartonella.rattimassiliensis.15908
Bartonella.elizabethae.Re6043vi
Bartonella.tribocorum.CIP.105476
Bartonella.grahamii.as4aup
Bartonella.doshiae.NCTC.12862
Bartonella.washoensis.085.0475
Bartonella.quintana.str..Toulouse
Bartonella.koehlerae.C.29
Bartonella.henselae.str..Houston.1

Figure 3.6f. Phylogenetic tree based on nucleotide similarity among tmk (thymidylate kinase) genes.

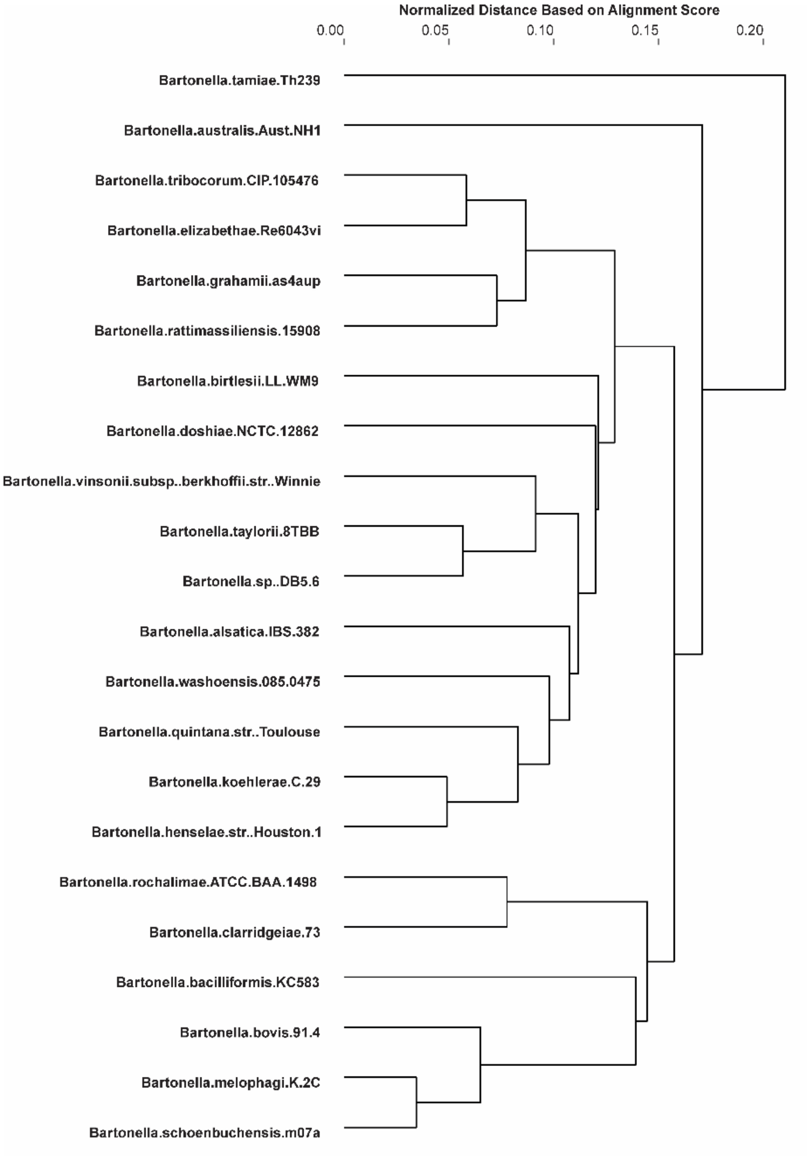

Normalized Distance Based on Alignment Score
0.00
0.05
0.10
0.15
0.20
Bartonella.tamiae.Th239
Bartonella.australis.Aust.NH1
Bartonella.tribocorum.CIP.105476
Bartonella.elizabethae.Re6043vi
Bartonella.grahamii.as4aup
Bartonella.rattimassiliensis.15908
Bartonella.birtlesii.LL.WM9
Bartonella.doshiae.NCTC.12862
Bartonella.vinsonii.subsp..berkhoffii.str..Winnie
Bartonella.taylorii.8TBB
Bartonella.sp..DB5.6
Bartonella.alsatica.IBS.382
Bartonella.washoensis.085.0475
Bartonella.quintana.str..Toulouse
Bartonella.koehlerae.C.29
Bartonella.henselae.str..Houston.1
Bartonella.rochalimae.ATCC.BAA.1498
Bartonella.clarridgeiae.73
Bartonella.bacilliformis.KC583
Bartonella.bovis.91.4
Bartonella.melophagi.K.2C
Bartonella.schoenbuchensis.m07a

Figure 3.6g. A consensus tree based on rpoC, aatA, dapD,rpsB and tmk genes.

Using the entire core genome identified using the clustering algorithm also allowed a more comprehensive consensus molecular-clock-based tree using all 714 core genes (Figure 3.7)

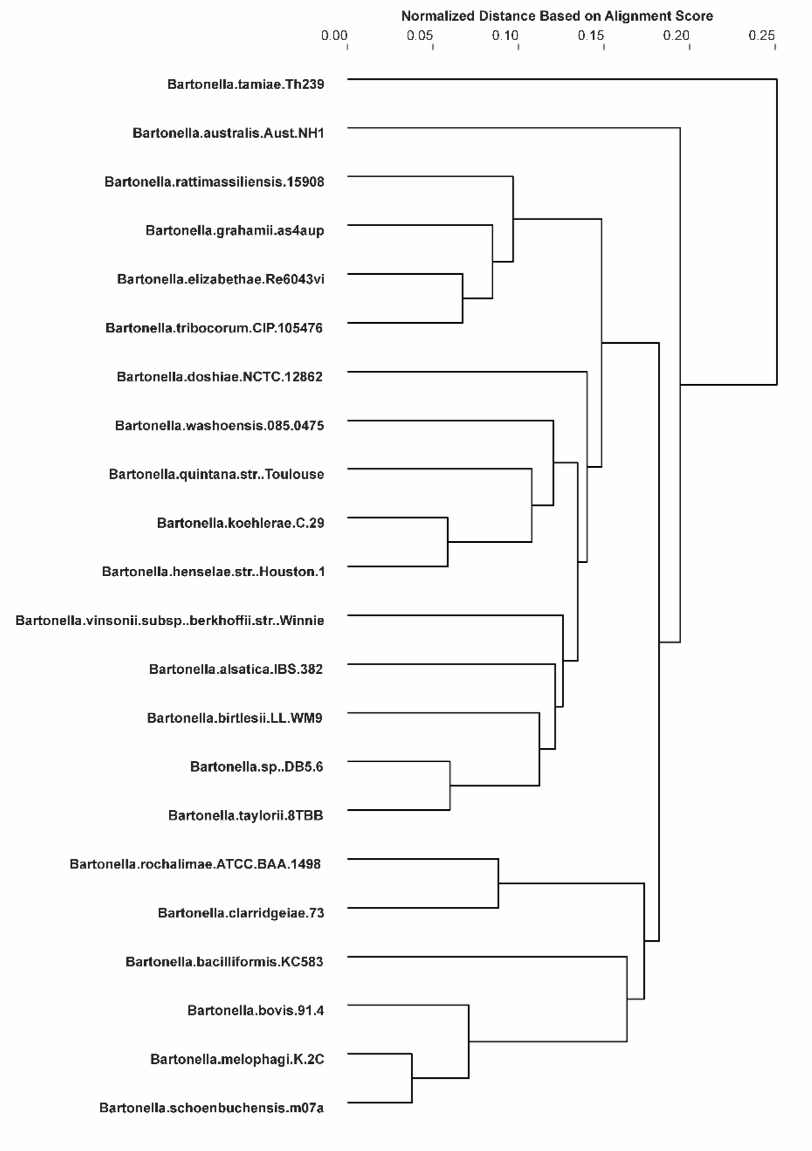

Normalized Distance Based on Alignment Score
0.00
0.05
0.10
0.15
0.20
0.25
Bartonella.tamiae.Th239
Bartonella.australis.Aust.NH1
Bartonella.rattimassiliensis.15908
Bartonella.grahamii.as4aup
Bartonella.elizabethae.Re6043vi
Bartonella.tribocorum.CIP.105476
Bartonella.doshiae.NCTC.12862
Bartonella.washoensis.085.0475
Bartonella.quintana.str..Toulouse
Bartonella.koehlerae.C.29
Bartonella.henselae.str..Houston.1
Bartonella.vinsonii.subsp..berkhoffii.str..Winnie
Bartonella.alsatica.IBS.382
Bartonella.birtlesii.LL.WM9
Bartonella.sp..DB5.6
Bartonella.taylorii.8TBB
Bartonella.rochalimae.ATCC.BAA.1498
Bartonella.clarridgeiae.73
Bartonella.bacilliformis.KC583
Bartonella.bovis.91.4
Bartonella.melophagi.K.2C
Bartonella.schoenbuchensis.m07a

Figure 3.7. Consensus tree based on 714 core genes from Bartonella spp.

The molecular clock tree reflects the time of separation of bacterial species in their evolution while the tree based on gene-profile similarities reflect functional similarity such as adaptation to the host and its microbiome, as well as various external physical (temperature, humidity, UV light exposure) and biochemical conditions (pH, NaCl concentration). These differences detected by gene profiles permit the reconstruction of the history of the species' evolution and adaptation including bottlenecks, host changes, and acquisitions/loss of pathways (which can also be a reflection of the surrounding microbiome history).

4. Core and pan genome estimation model for *Bartonella* spp.

Too few genomes used for analysis cannot identify new genes which may appear in a particular genus (pan genome) or identify which genes are conserved across all the species belonging to the genus (core genome). Figures 4.1 and 4.2 show the number of genes in common and the total number of genes present in groups containing 1, 2, 3, …, 22 *Bartonella* spp. genomes. It must be noted that the curve representing the pan genome (collector's curve [45]) in Figure 4.1 shows signs of saturation. The pan-genome size of *Bartonella* spp. is expected to approach a finite asymptotic value as number of sequenced *Bartonella* genomes increase.

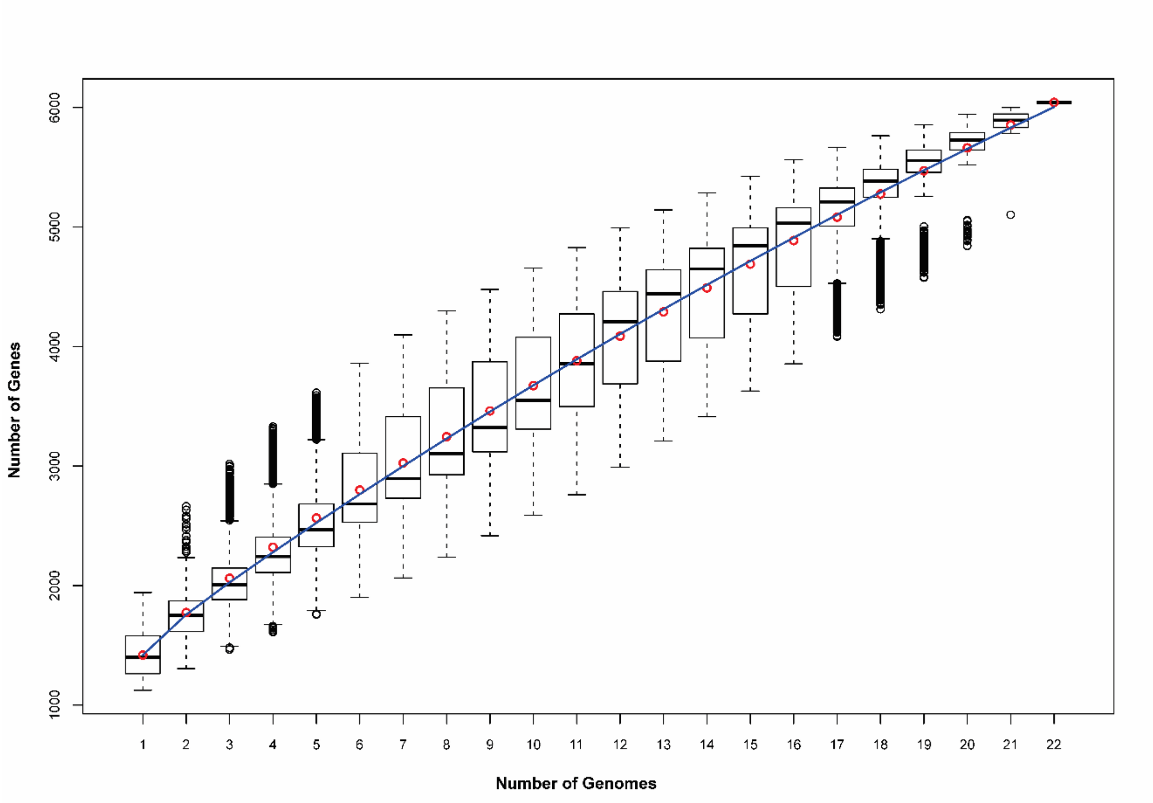


Figure 4.1. Box-and-Whisker plots of total number of genes found in varying size groups of Bartonella genomes. Each Box-and-Whisker plot represents the distribution of pan-genome size for all possible combinations of genomes within the group. The top and bottom whiskers represent the upper and lower 25% quartiles respectively – excluding outliers. The box represents the second and third quartiles (50%). The horizontal lines inside the boxes represent the group medians, red circles represent the group averages and the black circles represent the outliers. The blue curve represents the model fitting applied to group averages.

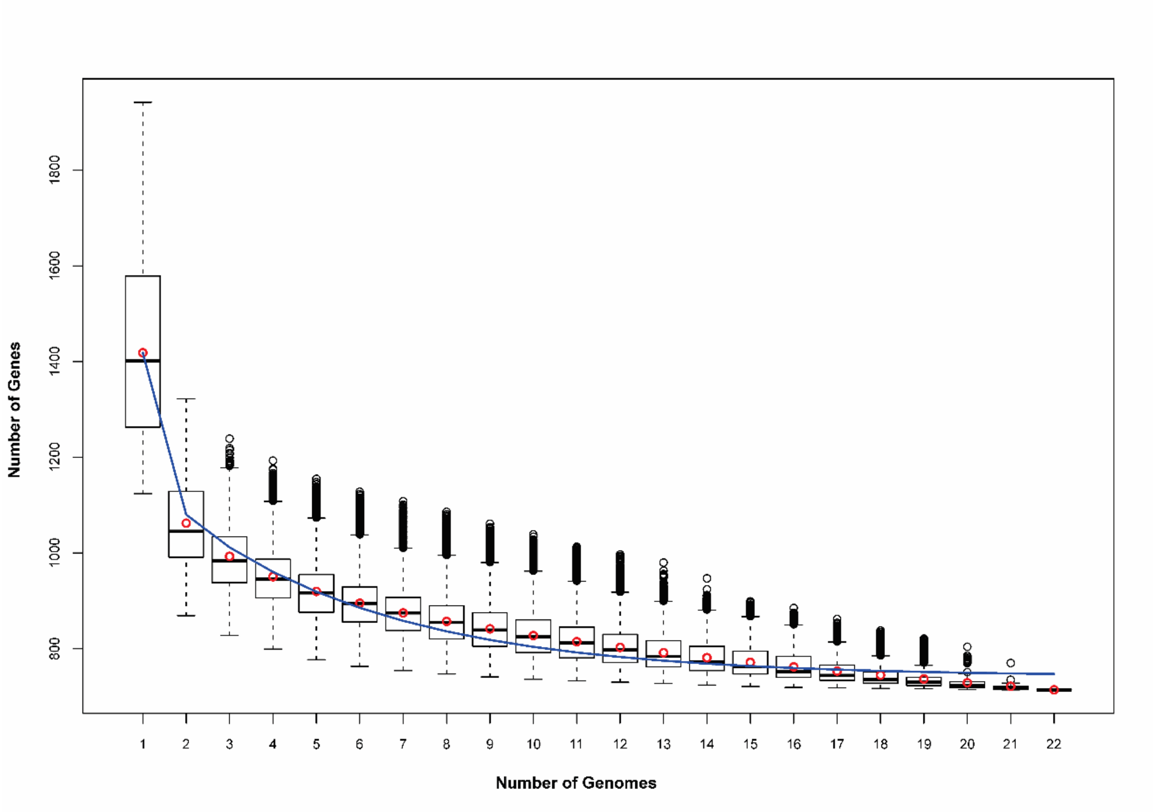


Figure 4.2. Box-and-Whisker plots of number of common genes found in varying size groups of Bartonella genomes. Each Box-and-Whisker plot represents the distribution of core-genome size for all possible combinations of genomes within the group. The top and bottom whiskers represent the upper and lower 25% quartiles respectively – excluding outliers. The box represents the second and third quartiles (50%). The horizontal lines inside the boxes represent the group medians, red circles represent the group averages and the black circles represent the outliers. The blue curve represents the model fitting applied to group averages.

Several studies [10]–[12], [8], [46]–[49] in the past 10 years have focused on pan genome analysis including its size estimation for many species of bacteria. Some of these studies also include development of the mathematical models to predict pan/core-genome sizes using various assumptions [10], [11], [50].

In order to fulfill the intuitive understanding of the core and pan genomes of bacteria, a mathematical model to predict them must include the following basic principles:

1- Finite size of the core genome.
2- Finite size of the pan genome.
3- Different genes must be present with different abundances/probabilities in the flexible (pan minus core) part of the pan genome.

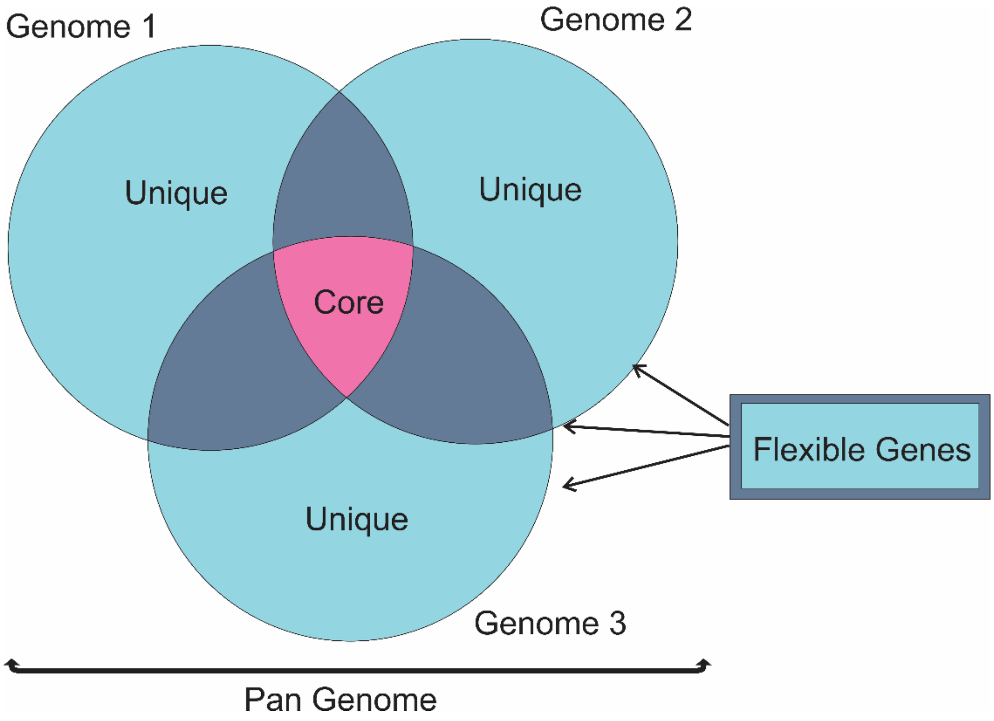


Figure 4.3. The illustration of the core and pan genome concepts.

Tettelin et al. 2005 [8] proposed using an exponential-decay function to estimate the number of new genes introduced into the pan genome of a species by the addition of a new genome, with the assumption that the pan genome is “open”. The assumption is made that the core-genome size approaches a finite value as more genomes are added, however, the function that describes the number of “new genes introduced to pan genome” is asymptotic to a non-zero value, indicating that the pan-genome size of bacteria is unbounded.

The assumption that the pan genome is unbounded violates the finite pan-genome size requirement. The existence of gene classes with different abundance/probabilities is also not considered in this model, violating the third requirement. Tettelin et al. [12] in 2008 proposed to use the Heaps’ law [51] model which allows for an “open” pan genome assumption. Based on model parameters, this approach can potentially produce an asymptotic pan genome curve which supports the finite pan genome assumption, however, it can also predict infinite growth of the pan genome. Similar to the previous study, the finite pan-genome size and different genes appearing with different abundance/probability requirements are violated.

In summary, the exponential-decay and power law models predict an infinite growth of the number of genes in the pan genome when the number of genomes under consideration increases. This assumption contradicts the common belief that the number of genes in pan genome is finite [7], [52]–[54] and the models are not adequate in describing multiple classes of genes appearing in different abundance/probability.

In 2007, Hogg et al. [54] proposed a finite pan genome model using a mixture of binomial distributions, introducing the concept of "*gene classes*". The model has been developed under the assumption that the pan genome contains classes of genes with equal probabilities but a different number of genes per class so the probability of a particular gene's appearance in a genome can be predicted based on predefined proportion of a particular gene class in a genome and the number of genes in this class. The model complies with all the principles described previously, however, the approach is not robust, so manually selected parameters can significantly affect results and the large degree of freedom can cause overfitting. Figure 4.4 shows the effect of choosing a number of parameters in pan genome-size estimation. Using this approach, Figure 4.5 shows these effects on core-genome-size estimation. The original study used seven predefined bins to create seven classes of genes. Each bin represented the probability to pick a particular gene from itself. For example, a gene class probability of 1.0 represents the gene class we refer to as the core genome. In order to measure the effects of the artificial creation of gene classes, we calculated the model's estimated core- and pan-genome sizes using a different number of bins.

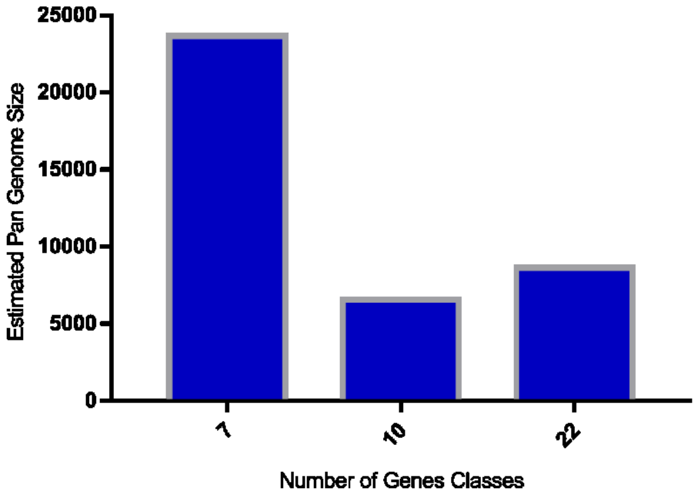


Figure 4.4. Effect of the number of bins used for pan-genome-size estimation. Hogg et al. [54] *binomial model*

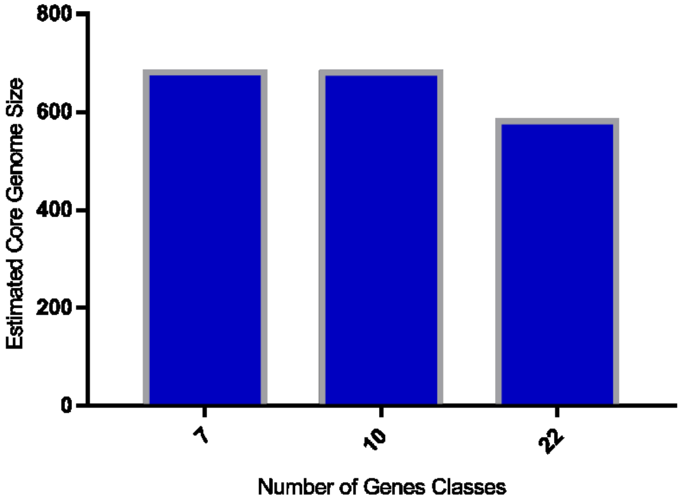


Figure 4.5. Effect of number of bins used for core-genome-size estimation. Hogg et al. [54] *binomial model*

It can be seen (Figures 4.4 and 4.5) that the number of bins chosen significantly affects the estimation.

4.1 Single Gene Pool Model

The following model has been proposed to resolve the prediction instability issues.

Let n be the average number of unique genes observed in an organism; m_{pan} be the total number of unique genes present in a finite pool of genes. Assume that m_{pan} consists of a fixed set of core genes present in every organism in the data set, m_{core}, and a flexible set of genes, m_{flex}. The number of unique genes expected to be found in the union of all organisms is

$$m_{pan} = m_{core} + m_{flex}$$

The number of unique genes expected to be found in a single organism, n, can be represented as m_{core} and some genes coming from of m_{flex}. The number of genes contributed to n by m_{flex} is:

$$n - m_{core}$$

Assuming that the appearance of a gene belonging to the flexible set of genes can be described as a random process, the probability of finding a gene in an organism that belongs to m_{flex} is then:

$$\frac{n - m_{core}}{m_{flex}}$$

The probability of finding a gene in k organisms simultaneously that belong to the flexible part is β^k and the number of genes expected to be in found in the intersection of k organisms that belong to the flexible part is:

$$m_{flex}(\frac{n - m_{core}}{m_{flex}})^k$$

The total number of genes expected to be found in the intersection of k organisms is:

$$f_{intersection,k} = m_{core} + m_{flex}(\frac{n - m_{core}}{m_{flex}})^k$$

Out of K, the total number of organisms, the number of unique combinations of k organisms C_k is:

$$C_k = \frac{K!}{(K-k)!\,k!}$$

The average number of unique of genes observed in the k^{th} combinatorial of K organisms in the intersection, $o_{intersection,k}$, is:

$$o_{intersection,k} = \frac{1}{C_K}\sum_{i=1}^{C_k} o_{intersection,i,k}$$

The relative error in estimating the number of genes found in the intersection is:

$$error_{intersection,k} = \left|\frac{o_{intersection,k} - f_{intersection,k}}{o_k}\right|$$

The probability of not finding a given gene from the flexible part in an organism is:

$$1 - \frac{n - m_{core}}{m_{flex}}$$

The probability of not finding the gene from the flexible part in any of the k organisms is:

$$(1-\frac{n-m_{core}}{m_{flex}})^k$$

The probability of finding the gene in at least one or more organisms is:

$$1-(1-\frac{n-m_{core}}{m_{flex}})^k$$

The number of unique genes expected to be observed in the union of k genomes is therefore:

$$f_{union,k} = m_{core} + m_{flex}(1 - \left(1-\frac{n-m_{core}}{m_{flex}}\right)^k)$$

The average number of unique of genes observed in the k^{th} combination of K organisms in the union, $o_{union,k}$, is:

$$o_{union,k} = \frac{1}{C_K}\sum_{i=1}^{C_k} o_{union,i,k}$$

The relative error in estimating the number of genes found in the union of k organisms is:

$$error_{union,k} = \left|\frac{o_{union,k} - f_{union,k}}{o_{union,k}}\right|$$

In order to estimate the size of the core and the pan genomes, we can minimize the total error for $m_{core}\ and\ m_{flex}$:

$$\min_{m_{flex}} \min_{m_{core}} \sum_{k=1}^{K} \left(error_{intersection,k} + error_{union,k}\right)$$

The number of possible combinations of values that are averaged for $o_{intersection,k}$ and $o_{union,k}$ are significantly different such as $C_1 = 22$ and $C_{11} = 705{,}432$ for the *Bartonella* dataset. The deviation in the averages calculated for values of k closer to $K/2$ is expected to be much smaller than values of k that are distant from $K/2$, therefore the error contribution of each step of k can be multiplied by a weight, $w_k = C_k$ to improve the fit.

$$\min_{m_{flex}} \min_{m_{core}} \sum_{k=1}^{K} w_k \left(error_{intersection,k} + error_{union,k} \right)$$

4.2 Mixture of Gene Pools Model

The single gene pool model based on the assumption that there is a single finite gene pool with an equal probability to pick genes cannot sufficiently describe highly diverse species. One can imagine a finite supply of gene pools of varying size and density to describe a diverse-species gene pool. It can be seen that the number of genes in the intersection of organisms is over-estimated and the union is under-estimated by a large difference for both data sets.

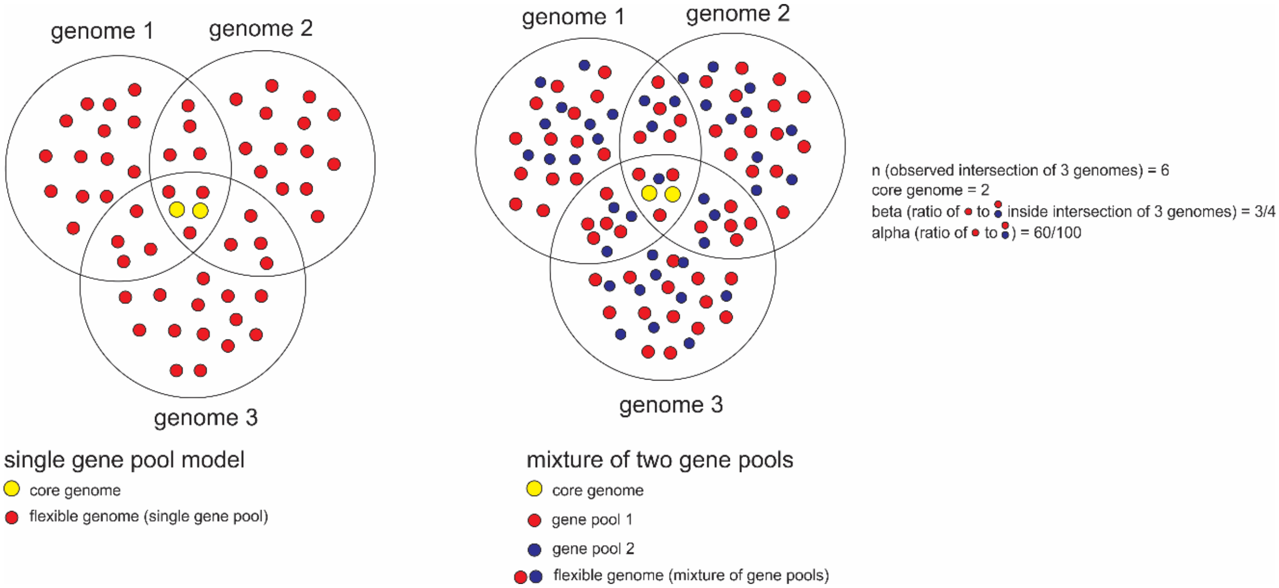

genome 1
genome 2
genome 3
single gene pool model
core genome
flexible genome (single gene pool)
genome 1
genome 2
genome 3
n (observed intersection of 3 genomes) = 6
core genome = 2
beta (ratio of ● to ●● inside intersection of 3 genomes) = 3/4
alpha (ratio of ● to ●●) = 60/100
mixture of two gene pools
core genome
gene pool 1
gene pool 2
flexible genome (mixture of gene pools)

Figure 4.6 Illustration of single and multiple gene pools models with different gene abundance/probability.

Assume that m_{flex} consists of smaller pools. Let α_i be the contribution ratio of the i^{th} pool out of N finite pools to m_{flex}:

$$m_{flex} = \sum_{i=1}^{N} \alpha_i m_{flex}$$

The pan-genome size can be defined as:

$$m_{pan} = m_{core} + \sum_{i=1}^{N} \alpha_i m_{flex}$$

If we assume that the significant portion of the observed genes can be represented by two independent pools:

$$\sum_{i=3}^{N} \alpha_i \ll \ \alpha_1 + \alpha_2$$

Since:

$$\sum_{i=1}^{N} \alpha_i = 1$$

Then:

$$\alpha_1 + \alpha_2 \;\cong 1$$

Similarly the proportional contribution of individual pools which make up m_{flex} to the average number of unique genes observed in an organism can also be defined as:

$$\sum_{i=3}^{N} \beta_i \ll \; \beta_1 + \beta_2$$

We can rewrite the model that estimates the unique number of genes found in the intersection of k organisms coming from the flexible part as:

$$m_{flex}\alpha_1 \left(\frac{\beta_1 (n - m_{core})}{m_{flex}\alpha_1} \right)^k + \; m_{flex}\alpha_2 \left(\frac{\beta_2 (n - m_{core})}{m_{flex}\alpha_2} \right)^k \; + m_{flex} \sum_{i=3}^{N} \alpha_i \left(\frac{\beta_i (n - m_{core})}{m_{flex}\alpha_i} \right)^k$$

Ignore the last term as $\alpha_1 + \alpha_2 \;\cong 1$

$$m_{flex}\alpha_1 \left(\frac{\beta_1 (n - m_{core})}{m_{flex}\alpha_1} \right)_1^k + \; m_{flex}\alpha_2 \left(\frac{\beta_2 (n - m_{core})}{m_{flex}\alpha_2} \right)_2^k$$

Replace $\alpha_2 \cong 1 - \alpha_1$, and $\beta_2 \cong 1 - \beta_1$, and add $m_{core}$to find the total number of unique genes expected to be observed in the intersection k organisms.

$$f_{intersection,k} = m_{core} + m_{flex}\alpha\left(\frac{\beta(n - m_{core})}{m_{flex}\alpha}\right)^k + m_{flex}(1-\alpha)\left(\frac{(1-\beta)(n - m_{core})}{m_{flex}(1-\alpha)}\right)^k$$

Similarly, the total number of unique genes expected to be observed in the union of k organisms is:

$$f_{union,k} = m_{core} + m_{flex}\alpha\left(1 - \left(1 - \frac{\beta(n - m_{core})}{m_{flex}\alpha}\right)^k\right) + m_{flex}(1-\alpha)\left(1 - \left(1 - \frac{(1-\beta)(n - m_{core})}{m_{flex}(1-\alpha)}\right)^k\right)$$

In order to estimate the size of the core and the pan genomes, using the same relative error calculation with weights, we can minimize the total error for $m_{core}, m_{flex}\ and\ \alpha\ and\ \beta$:

$$\min_{\beta}\min_{\alpha}\min_{m_{flex}}\min_{m_{core}}\sum_{k=1}^{K}\left(w_k\left(error_{intersection,k} + error_{union,k}\right)\right)$$

4.3 Bootstrapping and model fitting

Bootstrapping is a viable alternative to making inferences on the population mean of an unknown distribution for small sample sizes. In this study 50 samples of identical size (22) were generated by resampling the original dataset using equal probabilities for genome selection with replacement. The model was fit to each bootstrap sample and the averages of the model parameters were calculated.

The resulting core and pan-genome size estimates show variation due to *Bartonella* spp. present in the dataset that are significantly distant from each other.

Table 4.1. Averages of model parameters used in bootstrap fitting

m_{core}	788.53
m_{flex}	4656.23
α	0.91
β	0.48
estimated core genome size	788.53
estimated pan genome size	5444.75
core/pan ratio	0.14

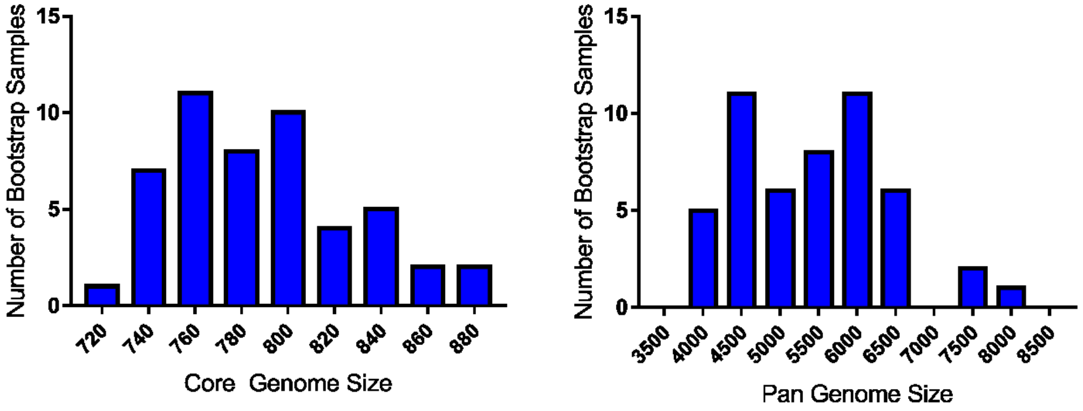
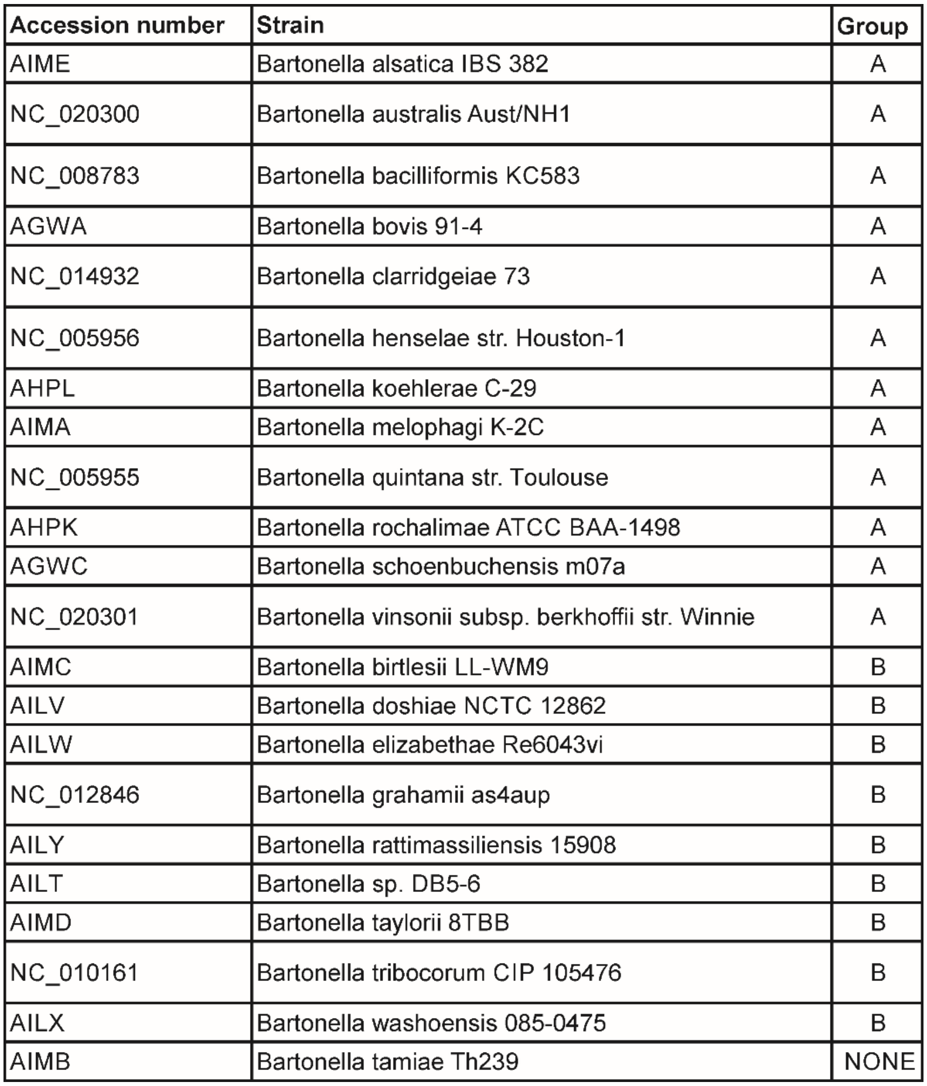


Figure 4.7 Distribution of model estimates for core and pan-genome size of bootstraps using all genomes

In order to observe the core/pan-genome size distribution of bootstraps created from functionally closer species and reduce the influence of outliers, separate bootstrap groups were created. We used the Jaccard index [41] as a score to determine the distance between genomes. In the gene-profile tree below there are two main groupings of *Bartonella* spp. falling under two main branches and one outlier (*Bartonella tamiae*). Two separate bootstrap sets were created using genomes labeled as group A and B (Table 4.2) excluding the outlier *Bartonella tamiae* genome. Each bootstrap group contains 50 datasets created using only the genomes allowed in the groups, with replacement.

Table 4.2 Bootstrap groups A and B



Accession number	Strain	Group
AIME	Bartonella alsatica IBS 382	A
NC_020300	Bartonella australis Aust/NH1	A
NC_008783	Bartonella bacilliformis KC583	A
AGWA	Bartonella bovis 91-4	A
NC_014932	Bartonella clarridgeiae 73	A
NC_005956	Bartonella henselae str. Houston-1	A
AHPL	Bartonella koehlerae C-29	A
AIMA	Bartonella melophagi K-2C	A
NC_005955	Bartonella quintana str. Toulouse	A
AHPK	Bartonella rochalimae ATCC BAA-1498	A
AGWC	Bartonella schoenbuchensis m07a	A
NC_020301	Bartonella vinsonii subsp. berkhoffii str. Winnie	A
AIMC	Bartonella birtlesii LL-WM9	B
AILV	Bartonella doshiae NCTC 12862	B
AILW	Bartonella elizabethae Re6043vi	B
NC_012846	Bartonella grahamii as4aup	B
AILY	Bartonella rattimassiliensis 15908	B
AILT	Bartonella sp. DB5-6	B
AIMD	Bartonella taylorii 8TBB	B
NC_010161	Bartonella tribocorum CIP 105476	B
AILX	Bartonella washoensis 085-0475	B
AIMB	Bartonella tamiae Th239	NONE

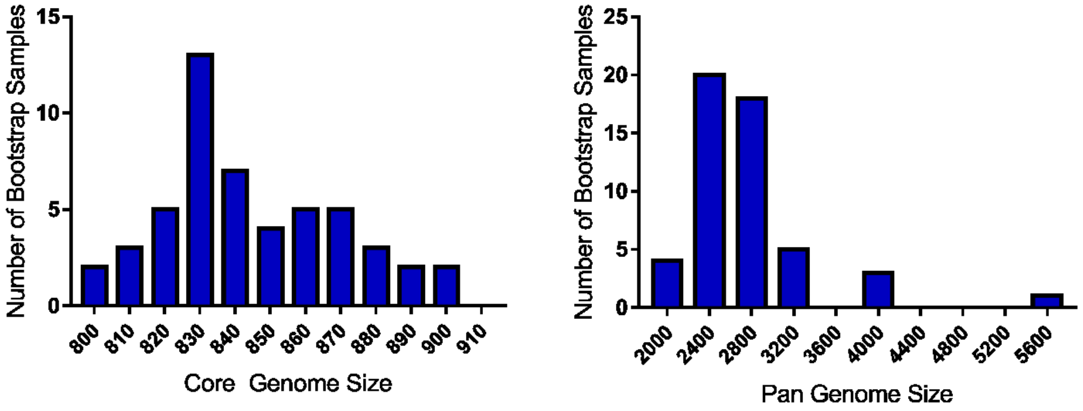


Figure 4.8. Distribution of model estimates for core- and pan-genome size for bootstraps created using genomes from group A

Table 4.3. Averages of model parameters used in bootstrap fitting for group A

m_{core}	844.19
m_{flex}	1921.91
α	0.84
β	0.48
estimated core genome size	844.19
estimated pan genome size	2766.1
core/pan ratio	0.31

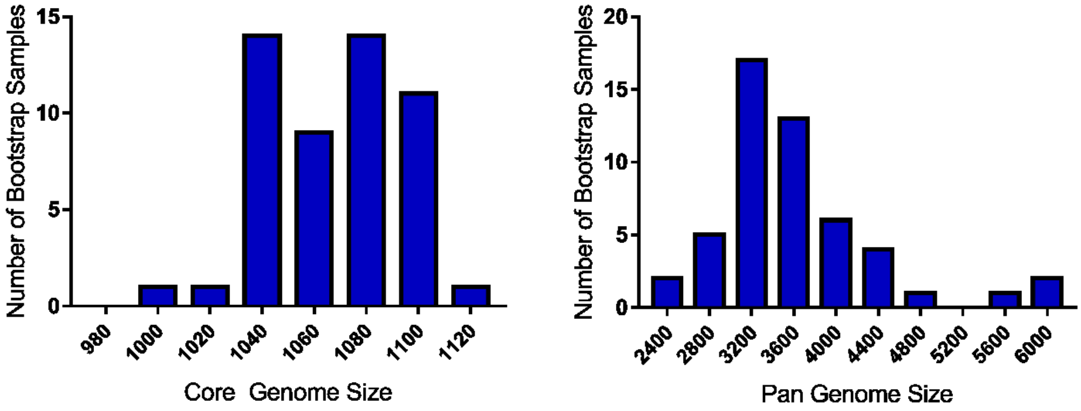


Figure 4.9. Distribution of model estimates for core- and pan-genome size for bootstraps created using genomes from group B

Table 4.4. Averages of model parameters used in bootstrap fitting for group B

m_{core}	1067.52
m_{flex}	2547.41
α	0.89
β	0.61
estimated core genome size	1067.52
estimated pan genome size	3614.92
core/pan ratio	0.30

There are 12 genomes in group A and 9 genomes in group B. The bootstraps generated using group A genomes are expected to share a smaller core genome than group B. Contrary to expectations, the average pan-genome size of bootstraps generated from group A is also smaller than group B. This observation may be attributed to the difference in number of genes found in these groups of genomes (average number of genes in group A is ~1,344 while group B is ~1,772).

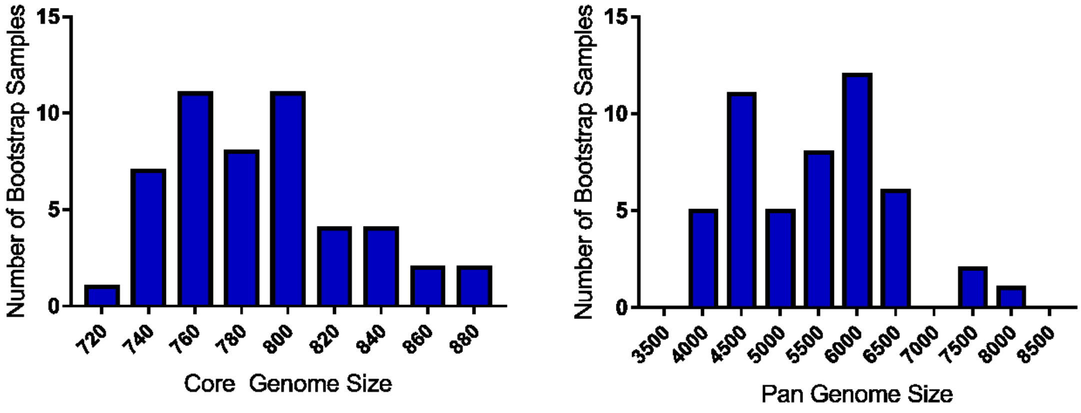


Figure 4.10. Distribution of model estimates for core- and pan-genome size for bootstraps using genomes from groups A and B (all genomes except Bartonella tamiae).

In order to observe the effects of the outlier organism (*Bartonella tamiae*), the same bootstrap process was repeated using genomes from both groups A and B excluding *Bartonella tamiae*. A lack of significant difference between averages shows that bootstrapping is effective in reducing the influence of outliers.

Table 4.5. Averages of model parameters used in bootstrap fitting for groups A and B combined (all genomes except Bartonella tamiae)

m_{core}	787.28
m_{flex}	4666.33
α	0.91
β	0.45
estimated core genome size	787.28
estimated pan genome size	5453.61
core/pan ratio	0.14

The proposed model appeared to be more robust, having less parameters and fulfill intuitive understanding of the core and pan genome. Bootstrapping showed that the model estimations are prone to sampling bias based on the selection of available genomes. This is especially true in practice as certain strains of a species are studied more often and therefore, complete genomes (with annotated genes) of a genus in a given database might predominantly contain only a few species of a genus.

Using this model, based on bootstrap averages (when all genomes are allowed), one can expect the total size of the core genome for 22 *Bartonella* genomes will be ~788 while the pan genome is expected to grow up to ~5,444 (Table 4.1). It also provides identification of the relative proportion of core (14%) and flexible (86%) genomes in *Bartonella* spp. The model can achieve the highest possible pan-genome size when all individual genomes are used without bootstrapping (with no repetitions allowed in the sampling). In this case, the core-genome size is expected to be smaller (~721) while the pan-genome size is expected to be significantly larger (~11,377) where the core genome only makes up ~6.4% of the pan genome. These findings reflect the wide range of biological, physical, and chemical conditions in which the bacteria can survive.

The parameters obtained from the model calculations using all genomes (upper bound) can be used for extrapolation in order to identify the minimum number of genomes required to capture at least 99% of all genes in *Bartonella* spp. pan genome (Figure 4.11). 99% of the pan genome (11,263 genes) can be captured using 186 *Bartonella* spp. genomes.

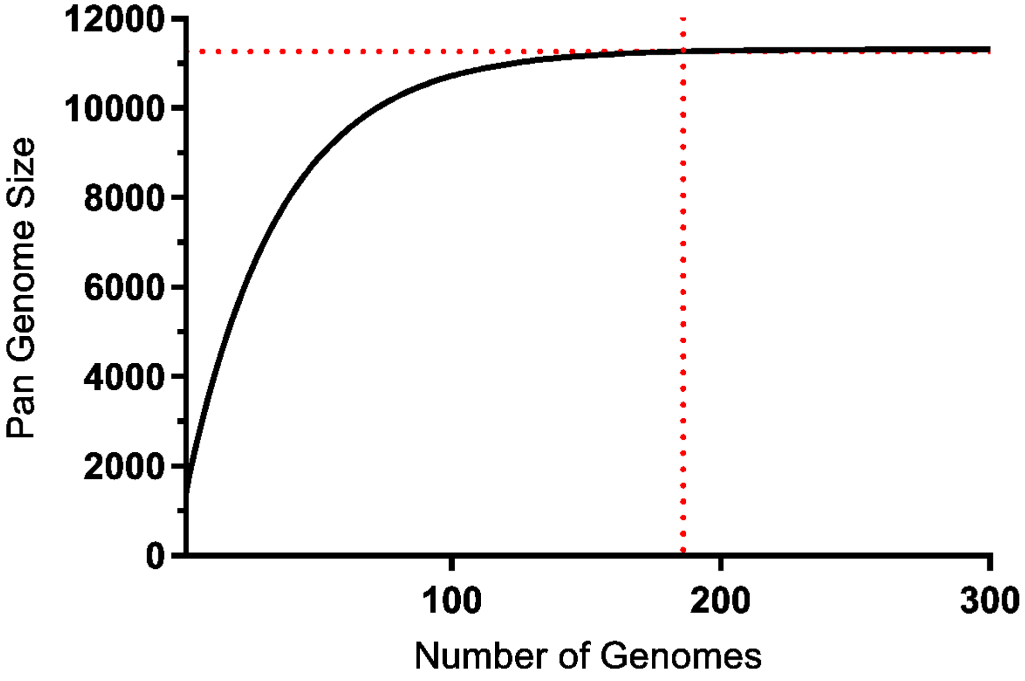


Figure 4.11 Extrapolation of Bartonella spp. pan-genome size using all available genomes

Conclusion and Future Work

Bacterial adaptation through gene loss and acquisition are the fundamental dynamic mechanisms that describe the nature of the core and pan genomes. Gene profiles reflect the functional adaptation of the organisms to their environment in contrast with the molecular clock based differentiation which reflects the time of separation from a common ancestor. The challenge in identifying gene profiles (which results in the identification of core and pan genomes as well) is the absence of appropriate clustering algorithms. In this work, we have developed methods to improve the quality and performance of gene clustering including heuristics free, novel sequence alignment algorithms that provide clustering of a large number of sequences significantly faster than traditional methods (a few days compared to months of computation). We demonstrated that the developed alignment algorithm (corridor restriction with logarithmic-sparse early-termination) requires significantly less number of calculations (84.72% less) than the original Hirschberg's algorithm. Such methods also enable the identification of appropriate similarity thresholds and form biologically meaningful cluster topology. The proposed mathematical model permits us to describe the core/pan genome of *Bartonella* spp. while the suggested methods allow creation of a functional similarity tree of *Bartonella* spp. based on gene profiles. Functional similarity-based trees, in contrast with molecular clock based similarity trees (using similarity of conserved gene(s)), can describe the environment and the role of organisms rather than time-based evolution (changes in nucleotide sequences of genes).

The developed approaches can be applied to other species of bacteria as well as extended to a higher level of taxonomy or used in comparison and characterization of microbial communities – previously only possible using heuristics. These new approaches can provide important insights and information to the long term evolution of the species and organization of microbial communities. The developed core/pan-genome-estimation method makes the ability to compare bacterial species based on their core/pan-genome size possible, which can provide additional information in understanding the functional diversity of species and their functional evolution.

References

[1] P. Raspor and D. Goranovic, “Biotechnological applications of acetic acid bacteria.,” *Crit. Rev. Biotechnol.*, vol. 28, no. 2, pp. 101–24, 2008.

[2] L. L. Barton and G. D. Fauque, “Biochemistry, physiology and biotechnology of sulfate-reducing bacteria.,” *Adv. Appl. Microbiol.*, vol. 68, pp. 41–98, 2009.

[3] E. Pennisi, “In industry, extremophiles begin to make their mark.,” *Science*, vol. 276, no. 5313, pp. 705–6, 1997.

[4] E. S. George Plopper, David Sharp, *Lewin’s CELLS 3rd Edition*. 2013.

[5] M. Juhas, “Horizontal gene transfer in human pathogens.,” *Crit. Rev. Microbiol.*, vol. 41, no. 1, pp. 101–8, 2015.

[6] N. Shterzer and I. Mizrahi, “The animal gut as a melting pot for horizontal gene transfer.,” *Can. J. Microbiol.*, vol. 61, no. 9, pp. 603–5, 2015.

[7] N. L. Hiller, B. Janto, J. S. Hogg, R. Boissy, S. Yu, E. Powell, R. Keefe, N. E. Ehrlich, K. Shen, J. Hayes, K. Barbadora, W. Klimke, D. Dernovoy, T. Tatusova, J. Parkhill, S. D. Bentley, J. C. Post, G. D. Ehrlich, and F. Z. Hu, “Comparative genomic analyses of seventeen *Streptococcus pneumoniae* strains: Insights into the *Pneumococcal* supragenome,” *J. Bacteriol.*, vol. 189, no. 22, pp. 8186–8195, 2007.

[8] H. Tettelin, V. Masignani, M. J. Cieslewicz, C. Donati, D. Medini, N. L. Ward, S. V Angiuoli, J. Crabtree, A. L. Jones, A. S. Durkin, R. T. Deboy, T.

M. Davidsen, M. Mora, M. Scarselli, I. Margarit y Ros, J. D. Peterson, C. R. Hauser, J. P. Sundaram, W. C. Nelson, R. Madupu, L. M. Brinkac, R. J. Dodson, M. J. Rosovitz, S. A. Sullivan, S. C. Daugherty, D. H. Haft, J. Selengut, M. L. Gwinn, L. Zhou, N. Zafar, H. Khouri, D. Radune, G. Dimitrov, K. Watkins, K. J. B. O'Connor, S. Smith, T. R. Utterback, O. White, C. E. Rubens, G. Grandi, L. C. Madoff, D. L. Kasper, J. L. Telford, M. R. Wessels, R. Rappuoli, and C. M. Fraser, "Genome analysis of multiple pathogenic isolates of *Streptococcus agalactiae*: implications for the microbial "pan-genome"," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 102, no. 39, pp. 13950–5, 2005.

[9] D. a. Rasko, M. J. Rosovitz, G. S. a. Myers, E. F. Mongodin, W. F. Fricke, P. Gajer, J. Crabtree, M. Sebaihia, N. R. Thomson, R. Chaudhuri, I. R. Henderson, V. Sperandio, and J. Ravel, "The pangenome structure of *Escherichia coli*: Comparative genomic analysis of *E. coli* commensal and pathogenic isolates," *J. Bacteriol.*, vol. 190, no. 20, pp. 6881–6893, 2008.

[10] G. Vernikos, D. Medini, D. R. Riley, and H. Tettelin, "Ten years of pan-genome analyses," *Curr. Opin. Microbiol.*, vol. 23, pp. 148–154, 2015.

[11] D. Medini, C. Donati, H. Tettelin, V. Masignani, and R. Rappuoli, "The microbial pan-genome," *Curr. Opin. Genet. Dev.*, vol. 15, no. 6, pp. 589–594, 2005.

[12] H. Tettelin, D. Riley, C. Cattuto, and D. Medini, "Comparative genomics: The bacterial pan-genome," *Curr. Opin. Microbiol.*, vol. 11, no. 5, pp. 472–

477, 2008.

[13] F. Baumdicker, W. R. Hess, and P. Pfaffelhuber, “The diversity of a distributed genome in bacterial populations,” *Ann. Appl. Probab.*, vol. 20, no. 5, pp. 1567–1606, 2010.

[14] O. Lukjancenko, T. M. Wassenaar, and D. W. Ussery, “Comparison of 61 sequenced *Escherichia coli* genomes.,” *Microb. Ecol.*, vol. 60, no. 4, pp. 708–20, 2010.

[15] R. C. Edgar, “Search and clustering orders of magnitude faster than BLAST.,” *Bioinformatics*, vol. 26, no. 19, pp. 2460–1, 2010.

[16] W. Li and A. Godzik, “Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences.,” *Bioinformatics*, vol. 22, no. 13, pp. 1658–9, 2006.

[17] T. A. Florin, T. E. Zaoutis, and L. B. Zaoutis, “Beyond cat scratch disease: Widening spectrum of *Bartonella henselae* infection.” *Pediatrics,* vol. 121, no. 5, pp. 1413-25, 2008.

[18] K. L. Karem, C. D. Paddock, and R. L. Regnery, “*Bartonella henselae*, *B. quintana*, and *B. bacilliformis*: Historical pathogens of emerging significance,” *Microbes Infect.*, vol. 2, no. 10, pp. 1193–1205, 2000.

[19] M. E. Ohl and D. H. Spach, “*Bartonella quintana* and urban trench fever,” *Clin. Infect. Dis.*, vol. 31, no. 1, pp. 131–135, 2000.

[20] C. Foucault, P. Brouqui, and D. Raoult, “*Bartonella quintana* characteristics

and clinical management,” *Emerg. Infect. Dis.*, vol. 12, no. 2, pp. 217–223, 2006.

[21] P. Brouqui, B. Lascola, V. Roux, and D. Raoult, “Chronic *Bartonella quintana* Bacteremia in homeless patients,” *N. Engl. J. Med.*, vol. 340, no. 3, pp. 184–189, 1999.

[22] C. Maguina, P. J. Garcia, E. Gotuzzo, L. Cordero, and D. H. Spach, “Bartonellosis (Carrión’s Disease) in the modern era,” *Clin. Infect. Dis.*, vol. 33, no. 6, pp. 772–779, 2001.

[23] Y. Regier, F. O Rourke, and V. A. J. Kempf, “*Bartonella* spp. - a chance to establish one health concepts in veterinary and human medicine.,” *Parasit. Vectors*, vol. 9, no. 1, p. 261, 2016.

[24] C. Silaghi, M. Pfeffer, D. Kiefer, M. Kiefer, and A. Obiegala, “*Bartonella*, rodents, fleas and ticks: A molecular field study on host-vector-pathogen associations in Saxony, Eastern Germany.,” *Microb. Ecol.*, published online ahead of print, 2016.

[25] Durbin, R., Eddy, S.R., Krogh, A. and Mitchison, G., *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.

[26] Donald E. Knuth, *The Art of Computer Programming: Volume 3: Sorting and Searching (2nd Edition)*. Addison Wesley Longman Publishing co., 1998.

[27] R. Xu and D. WunschII, “Survey of clustering algorithms,” *IEEE Trans. Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[28] 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, 1970.

[29] T. F. Smite and M. S. Waterman, “Identification of common molecular subsequences,” *Repr. from J . Mol. Biol. J. Mol. Bwl*, vol. 147, no. 147, pp. 195–197, 1981.

[30] D. S. Hirschberg, “A linear space algorithm for computing maximal common subsequences,” *Commun. ACM*, vol. 18, no. 6, pp. 341–343, 1975.

[31] W. Huang, D. M. Umbach, and L. Li, “Accurate anchoring alignment of divergent sequences.,” *Bioinformatics*, vol. 22, no. 1, pp. 29–34, 2006.

[32] A. Chakraborty and S. Bandyopadhyay, “FOGSAA: Fast optimal global sequence alignment algorithm.,” *Sci. Rep.*, vol. 3, p. 1746, 2013.

[33] F. Wilcoxon, “Individual comparisons by ranking methods,” *Biometrics Bull.*, vol. 1, no. 6, pp. 80–83, 1945.

[34] C. Pérez Vera, K. Aaltonen, T. Spillmann, O. Vapalahti, and T. Sironen, “Geographic distribution and molecular diversity of *Bartonella* spp. infections in moose (*Alces Alces*) in Finland,” *J. Wildl. Dis.*, vol. 52, no. 2, pp. 209–216, 2016.

[35] D. T. S. Hayman, K. D. McDonald, and M. Y. Kosoy, “Evolutionary history of rat-borne *Bartonella*: the importance of commensal rats in the dissemination of bacterial infections globally.,” *Ecol. Evol.*, vol. 3, no. 10, pp. 3195–203, 2013.

[36] S. Kumar, “Molecular clocks: Four decades of evolution.,” *Nat. Rev. Genet.*, vol. 6, no. 8, pp. 654–62, 2005.

[37] KD. Pruitt, G. Brown, T. Tatusova, and DR. Maglott, “NCBI reference sequences (RefSeq): A curated non-redundant sequence database of genomes, transcripts and proteins.,” *Nucleic Acids Res., vol. 35, database issue, pp. D61-5,* 2007.

[38] S. Paul, a. Bhardwaj, S. K. Bag, E. V. Sokurenko, and S. Chattopadhyay, “PanCoreGen - profiling, detecting, annotating protein-coding genes in microbial genomes,” *Genomics*, vol. 106, no. 6, pp. 367–372, 2015.

[39] J.-Q. Chen, Y. Wu, H. Yang, J. Bergelson, M. Kreitman, and D. Tian, “Variation in the ratio of nucleotide substitution and indel rates across genomes in mammals and bacteria.,” *Mol. Biol. Evol.*, vol. 26, no. 7, pp. 1523–31, 2009.

[40] M. A. Larkin, G. Blackshields, N. P. Brown, R. Chenna, P. A. McGettigan, H. McWilliam, F. Valentin, I. M. Wallace, A. Wilm, R. Lopez, J. D. Thompson, T. J. Gibson, and D. G. Higgins, “Clustal W and Clustal X version 2.0.,” *Bioinformatics*, vol. 23, no. 21, pp. 2947–8, 2007.

[41] M. Levandowsky and D. Winter, “Distance between sets,” *Nature*, vol. 234, no. 5323, pp. 34–35, 1971.

[42] G. R. R. Visweswaran, K. Leenhouts, M. van Roosmalen, J. Kok, and G. Buist, “Exploiting the peptidoglycan-binding motif, LysM, for medical and industrial applications.,” *Appl. Microbiol. Biotechnol.*, vol. 98, no. 10, pp. 4331–45, 2014.

[43] G. Buist, A. Steen, J. Kok, and O. P. Kuipers, “LysM, a widely distributed protein motif for binding to (peptido)glycans.,” *Mol. Microbiol.*, vol. 68, no. 4, pp. 838–47, 2008.

[44] M. Ventura, C. Canchaya, Z. Zhang, G. F. Fitzgerald, and D. van Sinderen, “Molecular characterization of hsp20, encoding a small heat shock protein of *bifidobacterium breve* UCC2003.,” *Appl. Environ. Microbiol.*, vol. 73, no. 14, pp. 4695–703, 2007.

[45] C. Deng, T. Daley, and A. D. Smith, “Applications of species accumulation curves in large-scale biological data analysis.,” *Quant. Biol. (Beijing, China)*, vol. 3, no. 3, pp. 135–144, 2015.

[46] T. Lefébure and M. J. Stanhope, “Evolution of the core and pan-genome of *Streptococcus*: Positive selection, recombination, and genome composition,” *Genome Biol.*, vol. 8, no. 5, p. R71, 2007.

[47] A. Jacobsen and R. S. Hendriksen, “The *Salmonella enterica* Pan-genome,” *Microb. Ecol.*, vol.62, no.3, pp. 487–504, 2011.

[48] P. Lapierre and J. P. Gogarten, “Estimating the size of the bacterial pan-genome,” *Trends Genet.*, vol. 25, no. 3, pp. 107–110, 2009.

[49] A. Mira, A. B. Martín-cuadrado, G. D. Auria, and F. Rodríguez-valera, “The bacterial pan-genome : A new paradigm in microbiology,” *Int. Microbiol.,* vol.13, no.2, pp. 45–57, 2010.

[50] F. Baumdicker, W. R. Hess, and P. Pfaffelhuber, “The infinitely many genes model for the distributed genome of bacteria,” *Genome Biol. Evol.,* vol. 4, no. 4, pp. 443–456, 2012.

[51] H. S. Heaps, *Information retrieval, computational and theoretical aspects*. Academic Press, 1978.

[52] T. Lefébure, P. D. P. Bitar, H. Suzuki, and M. J. Stanhope, “Evolutionary dynamics of complete *Campylobacter* pan-genomes and the bacterial species concept.,” *Genome Biol. Evol.*, vol. 2, pp. 646–55, 2010.

[53] R. Boissy, A. Ahmed, B. Janto, J. Earl, B. G. Hall, J. S. Hogg, G. D. Pusch, L. N. Hiller, E. Powell, J. Hayes, S. Yu, S. Kathju, P. Stoodley, J. C. Post, G. D. Ehrlich, F. Z. Hu, J. Kluytmans, A. van Belkum, H. Verbrugh, T. Coates, R. Bax, A. Coates, R. Daum, M. Stryjewski, H. Chambers, M. Otto, R. Klevens, M. Morrison, J. Nadle, S. Petit, K. Gershman, S. Ray, L. Harrison, R. Lynfield, G. Dumyati, J. Townes, A. Craig, E. Zell, G. Foshein, L. McDougal, R. Carey, S. Fridkin, F. Leonard, B. Markey, M. Nemati, K. Hermans, U. Lipinska, O. Denis, A. Deplano, M. Struelens, L. Devriese, F.

Pasmans, F. Haesebrouch, G. Ehrlich, G. Ehrlich, F. Hu, K. Shen, P. Stoodley, J. Post, F. Hu, G. Ehrlich, K. Shen, P. Antalis, J. Gladitz, S. Sayeed, A. Ahmed, S. Yu, J. Hayes, S. Johnson, B. Dice, R. Dopico, R. Keefe, B. Janto, W. Chong, J. Goodwin, R. Wadowsky, G. Erdos, J. Post, G. Ehrlich, F. Hu, K. Shen, J. Gladitz, P. Antalis, B. Dice, B. Janto, R. Keefe, J. Hayes, A. Ahmed, R. Dopico, N. Ehrlich, J. Jocz, L. Kropp, S. Yu, L. Nistico, D. Greenberg, K. Barbadora, R. Preston, J. Post, G. Ehrlich, F. Hu, K. Shen, S. Sayeed, P. Antalis, J. Gladitz, A. Ahmed, B. Dice, B. Janto, R. Dopico, R. Keefe, J. hayes, S. Johnson, S. Yu, N. Ehrlich, J. Jocz, L. Kropp, R. Wong, R. Wadowsky, M. Slifkin, R. Preston, G. Erdos, J. Post, G. Ehrlich, F. Hu, H. Tettelin, V. Masignani, M. Cieslewicz, C. Donati, D. Medini, N. Ward, S. Angiouli, J. Crabtree, A. Jones, A. Durkin, R. Deboy, T. Davidsen, M. Mora, M. Scarselli, yI M. Ros, J. Peterson, C. Hauser, J. Sundaram, W. Nelson, R. Madupu, L. Brinkac, R. Dodson, M. Rosowitz, S. Sullivan, S. Daugherty, D. Haft, J. Selengut, M. Gwinn, L. Zhou, N. Zafar, G. Ehrlich, A. Ahmed, J. Earl, N. Hiller, J. Costerton, P. Stoodley, J. Post, P. DeMeo, F. Hu, J. Hogg, F. Hu, B. Janto, R. Boissy, J. Hayes, R. Keefe, J. Post, G. Ehrlich, N. Hiller, B. Janto, J. Hogg, R. Boissy, S. Yu, E. Powell, R. Keefe, N. Ehrlich, K. Shen, J. Hayes, K. Barbadora, W. Klimke, D. Dernovoy, T. Tatusova, J. Parkhill, S. Bentley, J. Post, G. Ehrlich, F. Hu, B. Hall, G. Ehrlich, F. Hu, R. Aziz, D. Bartels, A. Best, M. DeJongh, T. Disz, R. Edwards, K. Formsma, S. Gerdes, E. Glass, M. Kubal, F. Meyer, G. Olsen, R. Olson, A. Osterman, R. Overbeek, L. McNeil, D. Paarmann, T. Paczian,

B. Parrello, G. Pusch, C. Reich, R. Stevens, O. Vassieva, V. Vonstein, A. Wilke, O. Zagnitko, M. Margulies, M. Egholm, W. Altman, S. Attiya, J. Bader, L. Bemben, J. Berka, M. Braverman, Y. Chen, Z. Chen, S. Dewell, L. Du, J. Fierro, X. Gomes, B. Godwin, W. He, S. Helgesen, C. Ho, G. Irzyk, S. Jando, M. Alenquer, T. Jarvie, K. Jirage, J. Kim, J. Knight, J. Lanza, J. Leamon, S. Lefkowitz, M. Lei, J. Li, C. Donati, N. Hiller, H. Tettelin, A. Muzzi, N. Croucher, S. Angiuoli, M. Oggioni, D. Riley, A. Covacci, S. Bentley, M. Kilian, G. Ehrlich, F. Hu, R. Rappuoli, E. Moxon, V. Masignani, C. Grasso, C. Lee, M. Mwangi, S. Wu, Y. Zhou, K. Sieradzki, H. de Lencastre, P. Richardson, D. Bruce, E. Rubin, E. Myers, E. Siggia, A. Tomasz, T. Baba, F. Takeuchi, M. Kuroda, H. Yuzawa, K. Aoki, A. Oguchi, Y. Nagai, N. Iwama, K. Asano, T. Naimi, H. Kuroda, L. Cui, K. Yamamoto, K. Hiramatsu, J. Fitzgerald, D. Sturdevant, S. Mackie, S. Gill, J. Musser, H. Tettelin, D. Riley, C. Cattuto, D. Medini, T. Lefebure, P. P. Bitar, H. Suzuki, M. Stanhope, M. Mussmann, F. Hu, M. Richter, D. de Beer, A. Preisler, B. Jørgensen, M. Huntemann, F. Glöckner, R. Amann, W. Koopman, R. Lasken, B. Janto, J. Hogg, P. Stoodley, R. Boissy, G. Ehrlich, T. Kwan, J. Liu, M. DuBow, P. Gros, J. Pelletier, A. Młynarczyk, G. Młynarczyk, J. Jeljaszewicz, L. Herron-Olson, J. Fitzgerald, J. Musser, V. Kapur, D. Halpern, H. Chiapello, S. Schbath, S. Robin, C. Hennequet-Antier, A. Gruss, M. El Karoui, D. Waldron, J. Lindsay, L. Weigel, D. Clewell, S. Gill, N. Clark, L. McDougal, S. Flannagan, J. Kolonay, J. Shetty, G. Killgore, F. Tenover, J. Evans, K. Dyke, C. Ubeda, E. Maiques, E. Knecht, I. Lasa, R.

Novick, J. Penadés, J. Michel, P. Yeh, R. Chait, R. Moellering, R. Kishony, S. Trindade, A. Sousa, K. Xavier, F. Dionisio, M. Ferreira, I. Gordo, I. Couto, H. de Lencastre, E. Severina, W. Kloos, J. Webster, H. de Lencastre, D. Oliveira, A. Tomasz, F. Slater, M. Bailey, A. Tett, S. Turner, P. Stoodley, L. Nistico, S. Johnson, L. Lasko, M. Baratz, V. Gahlot, G. Ehrlich, S. Kathju, and W. Pearson, "Comparative supragenomic analyses among the pathogens *Staphylococcus aureus*, *Streptococcus pneumoniae*, and *Haemophilus influenzae* using a modification of the finite supragenome model," *BMC Genomics*, vol. 12, no. 1, p. 187, 2011.

[54] J. S. Hogg, F. Z. Hu, B. Janto, R. Boissy, J. Hayes, R. Keefe, J. C. Post, and G. D. Ehrlich, "Characterization and modeling of the *Haemophilus influenzae* core and supragenomes based on the complete genomic sequences of Rd and 12 clinical nontypeable strains," *Genome Biol.,* vol. 8, no. 6, p. R103, 2007.