A MORE ACCURATE AND EFFICIENT WHOLE GENOME PHYLOGENY

P.Y. CHAN T.W. LAM S.M. YIU C.M. LIU

Department of Computer Science
The University of Hong Kong, Hong Kong
E-mail: {pychan, twlam, smyiu, cmliu}@cs.hku.hk

To reconstruct a phylogeny for a given set of species, most of the previous approaches are based on the similarity information derived from a subset of conserved regions (or genes) in the corresponding genomes. In some cases, the regions chosen may not reflect the evolutionary history of the species and may be too restricted to differentiate the species. It is generally believed that the inference could be more accurate if whole genomes are being considered. The best existing solution that makes use of complete genomes was proposed by Henz et al.[13] They can construct a phylogeny for 91 prokaryotic genomes in 170 CPU hours with an accuracy of about 70% (based on the measurement of non-trivial splits) while other approaches that use whole genomes can only deal with no more than 20 species. Note that Henz et al. measure the distance between the species using BLASTN which is not primarily designed for whole genome alignment. Also, their approach is not scalable, for example, it probably takes over 1000 CPU hours to construct a phylogeny for all 230 prokaryotic genomes published by NCBI. In addition, we found that non-trivial splits is only a rough indicator of the accuracy of the phylogeny. In this paper, we propose the followings.
(1) To evaluate the quality of a phylogeny with respect to a model answer, we suggest to use the concept of the maximum agreement subtree as it can capture the structure of the phylogeny.
(2) We propose to use whole genome alignment software (such as MUMmer) to measure the distances between the species and derive an efficient approach to generate these distances.
From the experiments on real data sets, we found that our approach is more accurate and more scalable than Henz et al.'s approach. We can construct a phylogenetic tree for the same set of 91 genomes with an accuracy more than 20% higher (with respect to both evaluation measures) in 2 CPU hours (more than 80 times faster than their approach). Also, our approach is scalable and can construct a phylogeny for 230 prokaryotic genomes with accuracy as high as 85% in only 9.5 CPU hours.

1. Introduction

Reconstructing a phylogeny for a given set of species is a well-known problem in computational biology. The resulting phylogeny can help researchers to understand the evolutionary history and relationship of the species. In the case of viruses, we may be able to identify the origin of the viruses so that precaution can be taken to avoid further spreading of the viruses. Therefore, an accurate and efficient reconstruction method is desirable.

Most of the previous approaches are based on a subset of conserved regions extracted from the corresponding genomes for the inference.[3, 5, 18, 28, 30] The distance between each pair of species is usually derived from the similarity of the selected regions. The accuracy of the produced phylogeny thus depends on the choice of these regions. Not surprisingly, there may be cases that these regions do not truly reflect the whole evolutionary history of the species. Different phylogenies may be obtained by selecting a different set of re-

gions. Or if only a small portion of the genomes is selected, there may be the problem of mutational saturation, that is, the selected regions are not powerful enough to differentiate the phylogenetic relations of some species. It is generally believed that the inference of phylogeny could be more accurate if the whole genomes are being used.[10, 13, 14]

However, there are two concerns for using the complete genomes: the scalability problem and the distance measure. To construct the phylogeny of a given set of species, we need to compute a distance for every pair of species. The amount of computation required grows quadratically with the number of species. Many previous attempts only deal with a small number of species (e.g., only nine and eleven genomes are considered by Hemiou et al.[14] and Fitz-Gibbon and House,[10] respectively). Also, how to derive a good distance measure for every pair of species is not completely resolved as most of the alignment tools are not designed for measuring the similarity (or distance) between two complete genomes.

The best existing solution along this direction was proposed by Henz et al.[13] They are able to construct a phylogeny for 91 prokaryotic genomes in 170 CPU[a] hours with an accuracy of about 70% when compared with phylogeny that is constructed using the taxonomy published by NCBI (we consider this as the *true* phylogeny). The accuracy measure used in their paper is based on the concept of *non-trivial splits*. Each internal edge in the phylogeny is called a non-trivial split. By deleting any of these edges, the species are separated into two groups. If there is a corresponding split in the true phylogeny, the split is considered to be good. The percentage of good splits is used as the accuracy measurement. In fact, using the percentage of good splits as the accuracy measure may not be a good indicator on the quality of the phylogeny. Figure 1 gives an example. The constructed phylogeny given in Figure 1(b) wrongly groups the whole Family B1 with Family A1 in the same subtree, and the Family B2 with Family A2 in another subtree. However, the accuracy based on non-trivial splits is as high as 92.3%. The problem is due to the fact that non-trivial splits do not explicitly capture the topology of the phylogenies.

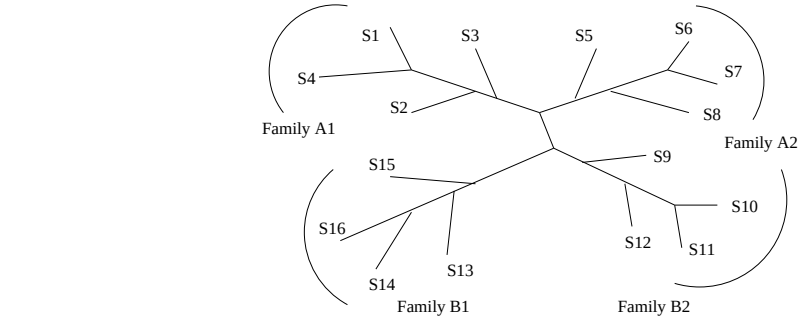
Moreover, their distance measure is based on the output of BLASTN which is not primarily designed for whole genome alignment. According to their approach, for each pair of genomes, BLASTN is executed to output a set of high-scoring local alignments. The total number of matched nucleotides from these alignments will be used as the similarity measure (and then the value is converted to a distance measure). However, there are examples where closer species may have a low score while two distant species may have a high score. For examples, the species Ralstonia solanacearum (Rs) and Neisseria meningitidis (Nm) should belong to the same group of beta-proteobacteria. On the other hand, the species Chlorobium tepidum (Ct) is from another family Chlorobi. However, based on the score from BLASTN, the distance of Ct from Nm is only 0.206 while the distance of Rs from Nm is about 3.86. That is why Nm and Ct are clustered together instead of Nm and Rs using Henz et al.'s approach (see Figure 2(a), for the mapping of the names of the species with the symbols, please refer to Figure 6). A similar example occurs to Treponema

[a] In their paper, they only report the CPU hours used without mentioning the actual running time which should be longer than the reported CPU hours. For our results, we will report both the CPU hours and the actual running time for comparison.

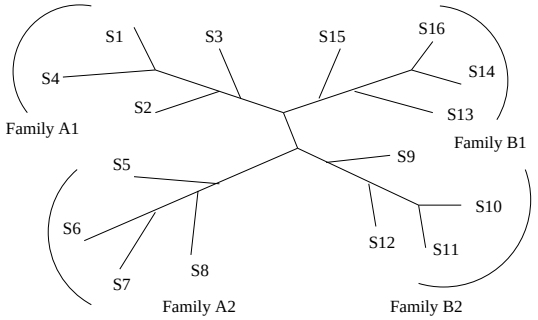
pallidum (Tp), Borrelia burgdorferi (Bb), and Clostridium perfringens (Cp).

We believe that the problem is due to the design of BLASTN which aims at locating all highly similar local alignments without considering the alignment of the whole genomes globally. Also, their approach is not scalable. It is estimated that to construct a phylogeny for all 230 prokaryotic genomes published by NCBI may take more than 1000 CPU hours which is not practical. To tackle these issues, in this paper, we propose the followings.

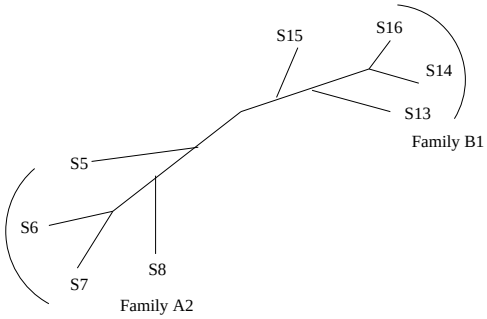
- To evaluate the quality of a phylogeny with respect to a true phylogeny, we suggest to use a well-known concept in the computer science community, called *maximum agreement subtree*,[6, 19] which captures the structure of the phylogeny and has been used for comparing the similarity of two given trees. In fact, the same concept has been used to compute a consensus tree given several different phylogenetic trees.[1, 15, 25] Roughly speaking, a maximum agreement subtree is defined as follows. From the constructed phylogeny, we select a maximum subset of species such that the resulting subtree based on these species should have the same topology (structure) as the resulting subtree derived using the same set of species in the true phylogeny. This subtree is called a maximum agreement subtree. The percentage of the selected species is used as the evaluation measurement.



(a) The Model Answer



(b) The Constructed Phylogeny



(c) A Maximum Agreement Subtree

Figure 1. Non-trivial Split may not be a good measure

Referring to the example in Figure 1, Figure (c) shows a maximum agreement subtree and the accuracy of the constructed phylogeny based on this new measure is 50% (contrary to the 92.3% based on non-trivial splits) which reflects the quality of the tree more appropriately. In Section 3, we will highlight the difference of two measures based on the output given in Henz et al.[13]

- For the distance measure, we propose to derive it from the output generated by the whole genome alignment software (such as MUMmer). Basically, we measure the number of matched nucleotides in the conserved regions reported by the software. We believe that the reported regions are more meaningful than the local alignments reported by BLASTN with respect to the comparison of two whole genomes.

  Most whole genome alignment software such as MUMmer are more efficient than BLASTN (note that they report different things). Yet a brute force approach to generate the distances for all pairs of genomes using MUMmer still requires a lot of computation. For example, it takes 9.5 CPU hours (i.e., 11.5 days of actual running time) to execute MUMmer for each pair of the 91 genomes tested by Henz et al. Although it is already much faster than Henz et al.'s approach, it is still not feasible for a larger set of species. So, we derive an efficient approach to speed up the generation of the the pairwise distances, enabling us to have a feasible solution for 230 genomes.

Table 1. Comparison of Two Approaches (Data Set I: 91 Prokaryotic Genomes)

	% of species in Max. Agreement Subtree	% of Good Splits	Running Time in CPU Hours (Actual Time in hours)	
Henz et al.'s Approach	60/91 = 65.9%	72.7% [13]	170 (Unknown)	
Our Approach (Using MUMmer)	81/91 = 89.0%	83/88 = 94.3%	Brute-force	9.5 (276)
			New Approach	2 (7)

Based on the experiments on real data sets, we found that our approach is more accurate and more scalable than Henz et al.'s approach (see Table 1). We can construct a phylogenetic tree on the same set of 91 genomes with an accuracy more than 20% higher (with respect to both evaluation measures) in 2 CPU hours (more than 80 times faster than their approach). The actual running time of our approach is only 7 hours. Our approach is scalable and can construct a phylogeny for 230 prokaryotic genomes with accuracy of 85% and 90% (with respect to our measure and the measure of good splits, respectively) in only 9.5 CPU hours (the actual running time is about 38 hours). In our experiments, we also tried a few different whole genome alignment tools, which all can provide a phylogeny with higher accuracy (details will be given in Section 4). It seems that the output provided by whole genome alignment software should provide a better distance measure than other software (such as BLASTN) that are not designed for whole genome alignment. As a remark, we have also tested two other whole genome alignment software (MSS [22] and Hybrid [4]) the accuracy of the predicted tree is more or less the same.

Organization of the paper: Section 2 discusses our approach, the distance measure we use, and how we speed up the whole procedure. We then describe the details of using maximum agreement subtree as our evaluation measure in Section 3. The experimental results will be presented in Section 4. Section 5 concludes the paper.

2. The Distance Measure and Our Approach

In this section, we describe our approach for generating the phylogenetic tree for a set of given species, in particular, the distance measure we use in the generation process. The following shows the framework of our approach.
Step 1: For each pair of species, perform the whole genome alignment using one of the selected software tools.
Step 2: Based on the output from the whole genome alignment software, we calculate a distance measure for each pair of species.
Step 3: Generate the phylogenetic tree using one of the distance-based phylogeny reconstruction software tools.

The Whole Genome Alignment Tools: The key difference between our approach and Henz et al.'s approach is that our distance measure is derived from the output given by software tools that are specially designed for locating conserved regions in the whole genome alignment. There are a number of software tools for whole genome alignment.[4, 7, 8, 29] They try to report all conserved regions of the given genomes. Most of these tools work as follows. They first identify a set of short substrings that are highly similar and unique in both genomes. These substrings are called *anchors*. These anchors provide a rough guideline on which parts of the genomes we should examine for conserved regions. It is obvious that not all anchors identified in the first step are useful as a lof of them may come from noise. The second step will consider these anchors based on different criteria and techniques (e.g. maximum common subsequence and clustering) so as to eliminate the noise and identify the conserved regions along the whole genomes. The set of anchors reported by the software is believed to be the markers for the conserved regions of the genomes. A common choice for anchors is the maximal substrings that are *exactly* matched and unique in the two genomes (called *MUM*). In this paper we use MUMs as our anchors for all experiments.

The Distance Measure: In order to show that the output from the whole genome alignment software tools is more appropriate for phylogenetic tree generation, we follow the idea of Henz et al.'s approach and use a straightforward distance measure. That is, we derive our measure from the total lengths of all the MUMs reported (that is, the selected anchors) by the software and normalize the value by the length of the shorter genomes. More precisely, we use the following distance measure.

$$\text{Distance Measure} = -\log_2\left(\frac{\text{Total MUM Length}}{\min\{\text{Lengths of Sequences}\}}\right)$$

In Henz et al.'s approach, instead of using the total MUM length, they use the total number of matched base pairs based on the set of high scored non-overlapping local alignments returned from BLASTN.

The Phylogeny Reconstruction Tools: In our research, we focus on distance-based phylogenetic reconstruction tools. Most of these software tools are based on two approaches: UPGMA[24] and Neighbor-Joining.[12, 20, 26] In this paper, our main purpose is not to evaluate the performance of different reconstruction tools. Therefore, based on the experimental results in Henz et al., BIONJ[12] performs the best among all the evaluated tools, so we also perform our experiments using BIONJ. Interested readers can refer to the PHYLIP package developed by Joe Felsenstein[9] for more information on different phylogenetic tree reconstruction software tools.

The Speed Up: In Step 1, for each pair of species, we have to identify a set of MUMs which requires the construction of a suffix tree for one of the species which also dominates the running time of the whole process (when using MUMmer). A brute-force approach would have to construct $O(n^2)$ suffix trees where n is the number of species. From our experiment on 91 prokaryotic genomes, the brute force approach will take about 9 CPU hours and 11.4 days of actual running time. Although it is faster than Henz et al.'s approach, it may not be feasible for a large set of species.

So, instead of constructing a suffix tree for each pair, we speed up the process as follows. We partition the species into groups of x species. We concatenate the genomic sequences of the species in each group and construct one suffix tree for each group, then for each sequence, we search against this suffix tree to locate MUMs for x pairs of species simultaneously. In other words, we avoid constructing the same suffix tree repeatedly as in the brute-force approach and also we speed up the searching process of MUMs by checking x pairs of species for MUMs in one round of searching. We are able to implement this approach in a PC with 4G memory by setting $x = 32$. The running time for the generation process decreases to 2 CPU hours (or 7 hours of actual running time) for 91 genomes.

From the viewpoint of theoretical analysis, we improve the time complexity from $O(mn^2)$ where m is the length of each genome to $O(mn)$ since the number of groups is small and can be considered as a constant in practice. We remark that the number of species (x) in each group should be calculated based on the available amount of memory and the sequence length of the species.

3. The Evaluation Measure

To evaluate the quality of a phylogenetic tree, we compare it to the phylogeny that is constructed using the taxonomy published in NCBI [b] (we call this the *true* phylogeny). One of the common concepts used for the comparison is the *non-trivial splits*.[13, 27] In this section, we formally define the measurement based on non-trivial splits and illustrate by a real example that this measure may not be a good indicator for the quality of the tree. Then, we propose to use the concept of *maximum agreement subtree*, a well-known concept in computer science community used for comparing the similarity of two given trees, to evaluate the quality of a predicted phylogeny.

[b] http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?CMD=search&DB=taxonomy

Non-Trivial Split: Given a phylogenetic tree, each internal edge, that is, the edge without a leaf (a species) attached to it is called a non-trivial split (we simply refer it as a split). For each split, if we delete the split, the tree will be partitioned into two connected components. All species will be divided into two sets according to which component that species belongs to. Intuitively, each split poses a classification on the species. If there is a corresponding split in the true phylogeny so that the species are partitioned exactly the same as that split. It means that the classification is correct and we call that split a *good* split. So, it is natural to define a measurement to evaluate the quality of the predicted tree as the percentage of the number of good splits out of the total number of splits in the predicted tree.

In Section 1 (Introduction), we provide an artificial example to illustrate that counting the percentage of good splits may not be a good indicator of the quality of the tree. In fact, splits do not explicitly capture the topology of the trees which is important in understanding the evolutionary history of the species. Also, some splits should be more important than the others. In particular, the split which separates a big family from another big family should be considered more important than a split which separates a species from the other species inside a subgroup. However, the measurement does not distinguish between these splits. In this section, we try to illustrate this problem using a real example.

Figure 2(a) shows the predicated phylogeny produced by Henz et al.'s approach on 91 prokaryotic genomes and Figure 2(b) is true phylogeny From the figures, one can see that the groups of alpha-proteobacteria, beta-proteobacteria, gamma-proteobacteria, and Spirochaete, are wrongly splitted into two or more subgroups attached to different parts of the phylogenetic tree. However, if we count the percentage of good splits, it is 72.7% which is a rather high score. It seems that this measurement may not be a good indicator.

The Maximum Agreement Subtree: The concept of *maximum agreement subtree* is not new in the computer science community and also, it has been used to reconcile different evolutionary trees and extract the maximum set of species such that the evolutionary relationships among these species are all agreed by these trees.[2, 16] Given two trees, T_1 and T_2, with leaves labelled by the same set of species, an *agreement subtree* is defined as follows. Let L_1 be a subset of species (leaves) in T_1. The subtree of T_1 induced by L_1 is an agreement subtree of T_1 and T_2 if this subtree is isomorphic to the subtree of T_2 induced by the same set of species L_1. Intuitively, if there is an agreement subtree induced by the subset L of species, it means that the evolutionary structure of these species are the same in both trees. If the size of L is the largest possible, then the corresponding agreement subtree is called a *maximum agreement subtree*.

Based on this idea, we derive a measure to evaluate the quality of the predicted tree by considering the largest possible size of L such that an agreement subtree exists. The percentage of the species that are selected in L is our proposed measure. Referring to Figure 2(a), if we use the percentage of species in the maximum agreement subtree as our quality measure (the selected species have been bolded in the figure), the evaluation score is 65.9% which we believe is a better score that reflects the quality of the tree.

Remark: In practice, the predicted tree is an unrooted binary tree, however, the true phylogeny is rooted and may not be a binary tree since the exact details of the evolutionary

history of the species in a subgroup may not be known. To compute the maximum agreement subtree (it is referred as the *maximum compatible subtree*), if there is a node in the true phylogeny with degree > 3, we allow it to be refined to a binary one by inserting artificial nodes so that deleting all these artificial nodes can get back the original subtree. In other words, we allow the predicted tree to have any evolutionary structure for these set of species. Similarly, the same applies to non-trivial splits. A non-trivial split in the predicted tree will be considered good if it corresponds to an artificial edge added because of the refinement process.

To compute the maximum compatible subtree is not trivial. Ganapathysaravanabavan and Warnow[11] provided a dynamic programming algorithm of $O(n^3 \times 2^{4d})$ time, where n is the number of species, for computing such a subtree for two unrooted trees with bounded degree $d+1$. However, the algorithm takes too long to compute (more than 30 minutes for 91 genomes and 172 hours for 230 genomes). In fact, the algorithm is a straight-forward extension of their algorithm for *rooted* trees. Many entries in the dynamic programming tables are computed more than once. We eliminate this redundancy by deriving a more efficient dynamic programming algorithm and the time complexity can be reduced by an $O(n)$ factor. Also, in our case, one of the trees has degree at most 3, so our algorithm runs in $O(n^2 \times 2^{2d})$ time. It only takes about 20 seconds and 45 minutes for 91 and 230 genomes, respectively.

4. Experimental Results

We have used two data sets for our experiments: Data Set I: 91 prokaryotic genomes that were used in the experiments of Henz et al.[13] Data Set II: all 230 prokaryotic genomes that are published in NCBI [c]. We use MUMmer [23] as the whole genome alignment software and work on the translated protein sequences of the genomes. For the phylogenetic tree reconstruction software, we use BIONJ.[21] The true phylogeny is derived from NCBI taxonomy in both data sets.

For both data sets, we evaluate our predicted phylogenetic tree using both measures. For Data Set I, from Table 1, we can see that our approach achieves an accuracy of more than 20% higher than Henz et al.'s approach in both measurements. Figure 3 shows the our predicted tree. For Data Set II, the accuracy of our predicted tree is 85.2% and 90.3% using the percentage of species in the maximum agreement subtree and good splits respectively. Figure 4 and 5 show the true phylogeny and our predicted tree for Data Set II. To conclude, our approach provides a more accurate method to predict phylogenetic tree.

For running time, our approach only requires 2 CPU hours (or 7 hours of actual running time) for Data Set I and 9.5 CPU hours (or 38 hours of actual running time) for Data Set II. Our approach is much faster (more than 80 times) than Henz et al.'s approach and in fact, their approach is not feasible for Data Set II as the estimated computation required will be more than 1000 CPU hours. So, our approach is more scalable than their approach.

Remark: We have also tried some other whole genome alignment software tools: MSS[22]

[c] http://www.ncbi.nlm.nih.gov/genomes/lproks.cgi

and the hybrid approach[4, 22] that combines MaxMinCluster[29] and MSS. We use the same proposed distance measure to construct the distance in all cases. For both measures and data sets, our approach (no matter which software tool is used) is able to produce phylogenies with higher quality (18% higher using MSS and more than 20% for hybrid). It illustrates that the output from the whole genome alignment tools is useful in constructing phylogenetic trees. On the other hand, MSS is quite intensive in computation. So, it takes longer time if we use these two software tools (for Data Set I, about 30 and 95 CPU hours are required, respectively, for MSS and hybrid) although it is already fast than Henz et al.'s approach.

5. Conclusion

In this paper, we study the problem of using whole genomes to reconstruct a phylogeny for a given set of species. We propose to derive the distance from the output reported by software tools that are specially designed for whole genome alignment. Experiments show that our proposed approach outperforms the existing approaches that do not make use of whole genome alignment to derive the distance measure and is able to infer a phylogenetic tree with a much higher accuracy. Moreover, our approach is more scalable and can be used to reconstruct a phylogeny for 230 prokaryotic genomes. Regarding the evaluation of a phylogeny, we point out that the evaluation based on non-trivial splits may not be a good indicator and we propose to use the concept of maximum agreement subtree which can also capture the structure of the tree.

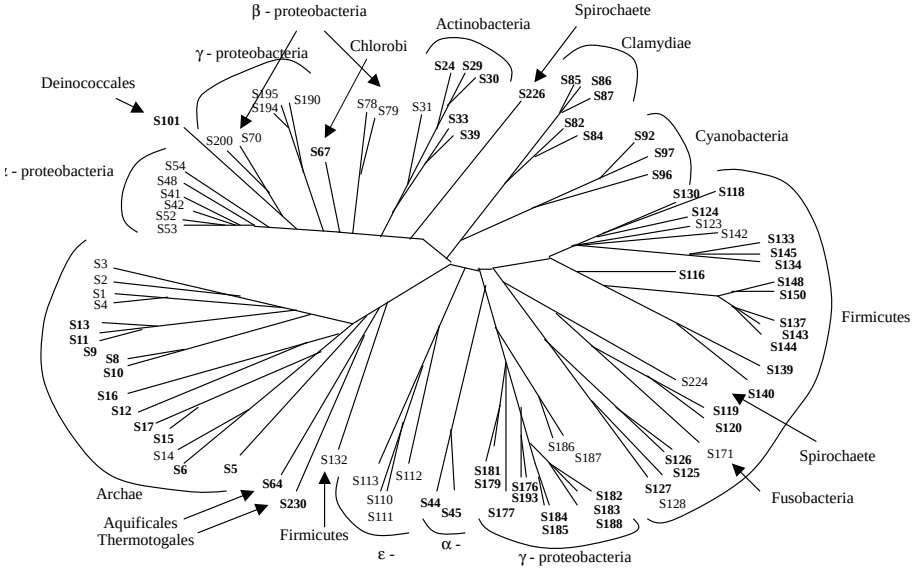
For further work, we will try to apply the same approach to the eurokaryotic genomes and try to derive other distance measures, for example, measures that can capture the number of mutations, in order to further improve the accuracy of the predicted phylogeny. A detailed study on the measures and the related issues, such as the normalization[17] would be carried out.

References

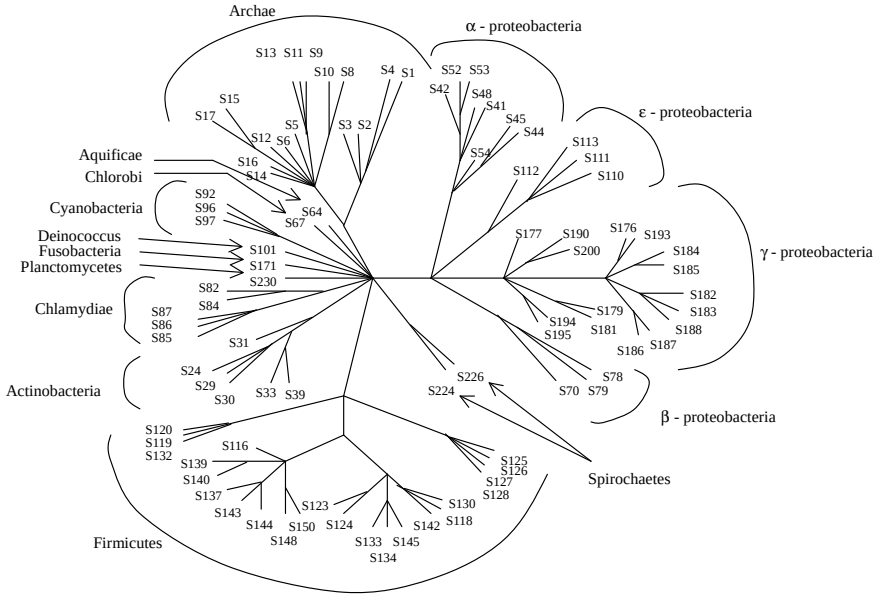
1. A. Amir and D. Kesselman. Maximum agreement subtree in a set of evolutionary trees - metrics and efficient algorithms. In *Proceedings of the 35th IEEE FOCS*, pages 758–769, 1994.
2. Amidhood Amir and Dmitry Keselman. Maximum agreement subtree in a set of evolutionary trees: Metrics and efficient algorithms. *SIAM Journal on Computing*, 26(6):1656–1669, 1997.
3. D.K. Bideshi, Y. Bigot, and B.A. Federici. Molecular characterization and phylogenetic analysis of the *harrisina brillians* granulovirus granulin gene. *Arch. Virol.*, 145:1933–1945, 2000.
4. HL Chan, TW Lam, WK Sung, Prudence WH Wong, and SM Yiu. The mutation subsequence problem and locating conserved genes. *Bioinformatics*, 21(10):2271–2278, 2005. A preliminary version appears in the Proceedings of the IEEE Fourth Symposium on Bioinformatics and Bioengineering (BIBE 2004).
5. X. Chen, W.F.J. Ijkel, C. Dominy, P. Zanotto, Y. Hashimoto, O. Faktor, T. Hayakawa, C.-H. Wang, A. Prekumar, S. Mathavan, P.J. Krell, Z. Hu, and J.M. Vlak. Identification, sequence analysis and phylogeny of the *lef*-2 gene of *helicoverpa armigera* single-nucleocapsid baculovirus. *Virus Research*, 65:21–32, 2001.
6. R. Cole, M. Farach, R. Hariharan, T. Przytycka, and M. Thorup. An $o(n \log n)$ algorithm for the maximum agreement subtree problem for binary trees. *SIAM Journal on Computing*, 30(5):1385–1404, 2000.

7. A.L. Delcher, S. Kasif, R. D. Fleischmann, J. Peterson, O. White, and S.L. Salzberg. Alignment of whole genomes. *Nucleic Acids Research*, 27(11):2369–2376, 1999.
8. A.L. Delcher, A. Phillippy, J. Carlton, and S.L. Salzberg. Fast algorithms for large-scale genome alignment and comparison. *Nucleic Acids Research*, 30(11):2478–2483, 2002.
9. J. Felsenstein. PHYLIP - phylogeny inference package (version 3.2). *Cladistics*, 5:164–166, 1989. http://evolution.genetics.washington.edu/phylip.html.
10. S.T. Fitz-Gibbon and C.H. House. Whole genome-based phylogenetic analysis of free living microorganisms. *Nucleic Acids Research*, 27:4218–4222, 1999.
11. Ganeshkumar Ganapathysaravanabavan and Tandy Warnow. Finding a maximum compatible tree for a bounded number of trees with bounded degree is solved in polynomial time. In *Proceedings of WABI 2001*, pages 156–163, 2001.
12. O. Gascuel. BIONJ: An improved version of the NJ algorithm based on a simple model of sequence data. *Mol. Biol. Evol.*, 14:685–695, 1997.
13. Stafan R. Henz, Daniel H. Huson, Alexander F. Auch, Kay Nieselt-Struwe, and Stephen C. Schuster. Whole-genome prokaryotic phylogeny. *Bioinformatics*, 21(10):2329–2335, 2005.
14. E.A. Herniou, T. Luque, X. Chen, J.M. Vlak, D. Winstanley, J.S. Cory, and D.R. O'Reilly. Use of whole genome sequence data to infer baculovirus phylogeny. *Journal of Virology*, 75(17):8117–8126, 2001.
15. M.-Y. Kao, T.W. Lam, T. Przytycka, W.K. Sung, and H.F. Ting. Efficient algorithms for comparing unrooted evolutionary trees. In *Proceedings of STOC*, pages 54–65, 1997.
16. Chuan-Min Lee, Ling-Ju Hung, Maw-Shang Chang, and Chuan-Yi Tang. An improved algorithm for the maximum agreement subtree problem. In *Proceedings of the 4th IEEE Symposium on Bioinformatics and Bioengineering (BIBE'04)*, page 533, 2004.
17. M. Li, X. Chen, X. Li, B. Ma, and P.M.B. Vitányi. The similarity metric. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 863–872, 2003.
18. G.J. Olsen, C.R. Woese, and R. Overbeek. The winds of (evolutionary) change: Breathing new life into microbiology. *J. Bact.*, 176:1–6, 1994.
19. T. Przytzcka. Sparse dynamic programming for maximum agreement subtree problem. In *Mathematical Hierarchies and Biology, DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 1997.
20. N. Saitou and M. Nei. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.*, 4:406–425, 1987.
21. The BIONJ Web Site. http://bioweb.pasteur.fr/seqanal/interfaces/bionj.html.
22. The MSS Web Site, 2004. http://www.cs.hku.hk/m̃ss.
23. The MUMMER Web Site, 2003. http://www.tigr.org/software/mummer/.
24. R.R. Sokal and C.D. Michener. A statistical method for evaluating systematic relationships. *University of Kansas Scientific Bulletin*, 28:1409–1438, 1958.
25. M. Steel and T. Warnow. Kaikoura tree theorems: Computing the maximum agreement subtree. *Information Processing Letters*, 48:77–82, 1993.
26. J.A. Studier and K.J. Keppler. A note on the neighbour-joining algorithm of saitou and nei. *Mol. Biol. Evol.*, 5:729–731, 1988.
27. L.-S. Wang and T. Warnow. Estimating true evolutionary distances between genomes. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC 2001)*, pages 637–646, 2001.
28. C.R. Woese. Bacterial evolution. *Microbiol. Rev.*, 51:221–272, 1987.
29. Prudence W.H. Wong, T. W. Lam, N. Lu, H. F. Ting, and S. M. Yiu. An efficient algorithm for optimizing whole genome alignment with noise. *Bioinformatics*, 20(16):2676–2684, 2004.
30. P.M.D. Zanotto, B.D. Kessing, and J.E. Maruniak. Phylogenetic interrelationships among baculoviruses: Evolutionary rates and host associations. *J. Invertebr. Pathol.*, 62:147–164, 1993.

Appendix



(a) The Best Phylogenetic tree for 91 species produced by Henz et al.'s Approach



(b) The True Phylogeny based on NCBI taxonomy (June, 2005) for 91 species

Figure 2. Phylogenetic trees produced by Henz et al.'s Approach and NCBI Taxonomy for 91 Species

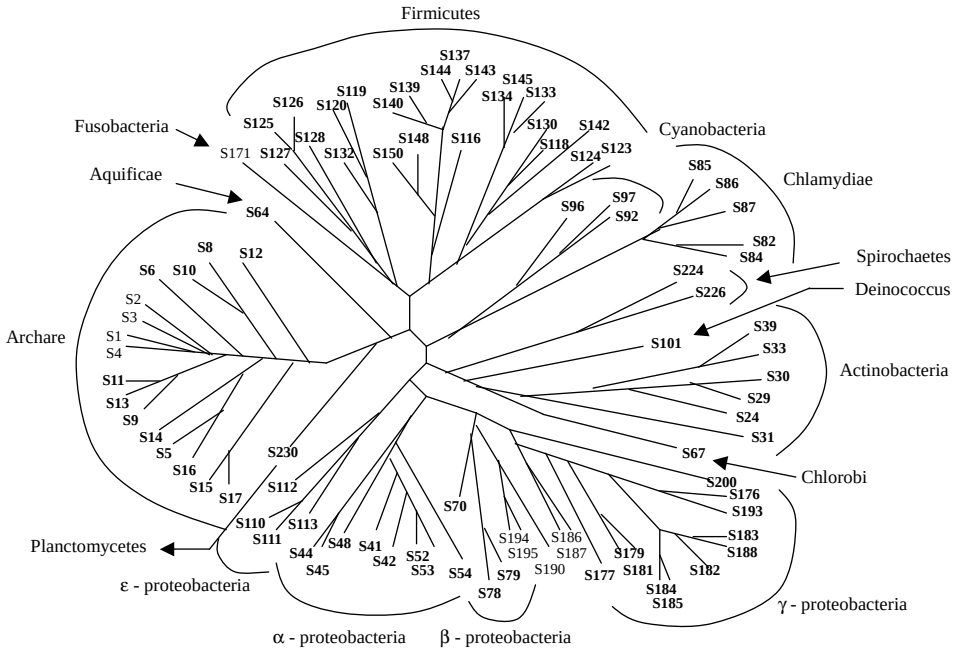


Figure 3. Phylogenetic trees produced by Our approach for 91 Species

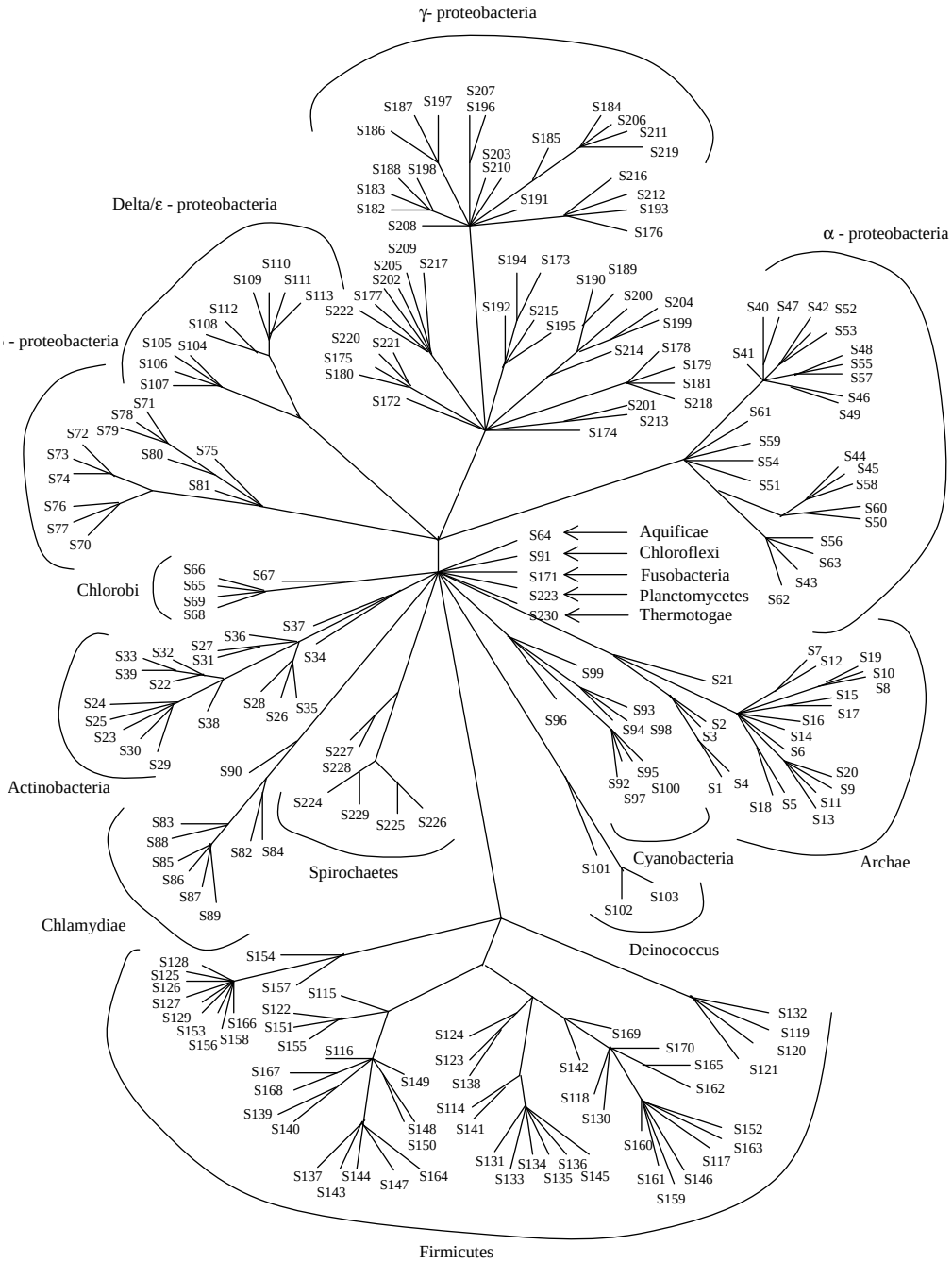


Figure 4. The True Phylogeny based on NCBI Taxonomy (June 2005) for 230 Species

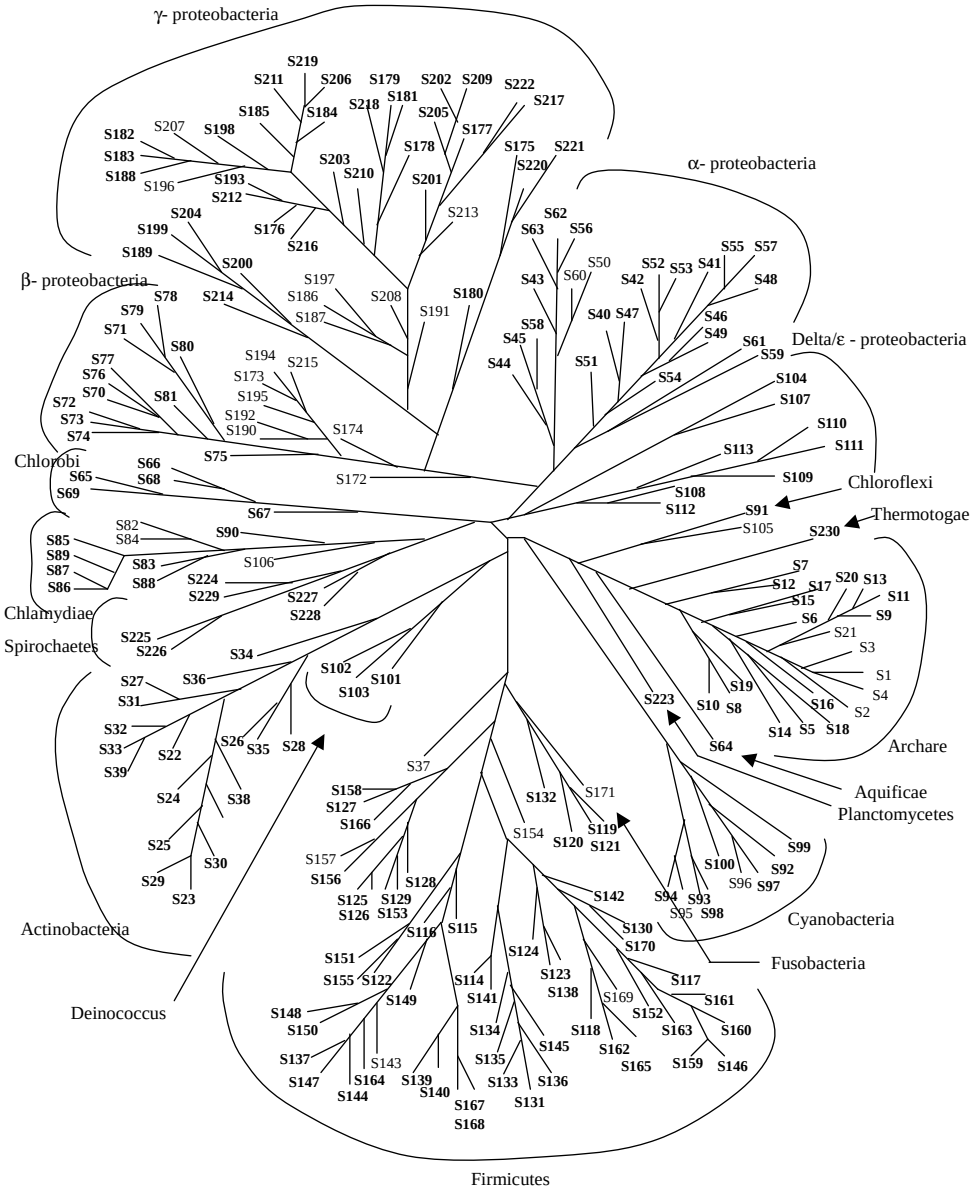


Figure 5. The Phylogenetic tree produced by Our Approach for 230 Species

S1 Sulfolobus solfataricus P2
S2 Pyrobaculum aerophilum str. IM2
S3 Aeropyrum pernix K1
S4 Sulfolobus tokodaii str. 7
S5 Methanocaldococcus jannaschii DSM 2661
S6 Archaeoglobus fulgidus DSM 4304
S7 Haloarcula marismortui ATCC 43049
S8 Thermoplasma acidophilum DSM 1728
S9 Pyrococcus abyssi GE5
S10 Thermoplasma volcanium GSS1
S11 Pyrococcus horikoshii OT3
S12 Halobacterium sp. NRC-1
S13 Pyrococcus furiosus DSM 3638
S14 Methanobacterium_thermoautotrophicum
S15 Methanosarcina acetivorans C2A
S16 Methanopyrus kandleri AV19
S17 Methanosarcina mazei Go1
S18 Methanococcus maripaludis S2
S19 Picrophilus torridus DSM 9790
S20 Thermococcus kodakarensis KOD1
S21 Nanoarchaeum equitans Kin4-M
S22 Corynebacterium diphtheriae NCTC 13129
S23 Mycobacterium bovis AF2122/97
S24 Mycobacterium leprae TN
S25 Mycobacterium avium subsp. paratuberculosis str. k10
S26 Tropheryma whipplei str. Twist
S27 Streptomyces avermitilis MA-4680
S28 Leifsonia xyli subsp. xyli str. CTCB07
S29 Mycobacterium tuberculosis CDC1551
S30 Mycobacterium tuberculosis H37Rv
S31 Streptomyces coelicolor A3(2)
S32 Corynebacterium efficiens YS-314
S33 Corynebacterium glutamicum ATCC 13032
S34 Bifidobacterium longum NCC2705
S35 Tropheryma whipplei TW08/27
S36 Propionibacterium acnes KPA171202
S37 Symbiobacterium thermophilum IAM 14863
S38 Nocardia farcinica IFM 10152
S39 Corynebacterium glutamicum ATCC 13032
S40 Bradyrhizobium japonicum USDA 110
S41 Mesorhizobium loti MAFF303099
S42 Sinorhizobium meliloti 1021
S43 Anaplasma marginale str. St. Maries
S44 Rickettsia conorii str. Malish 7
S45 Rickettsia prowazekii str. Madrid E
S46 Bartonella quintana str. Toulouse
S47 Rhodopseudomonas palustris CGA009
S48 Brucella melitensis 16M
S49 Bartonella henselae str. Houston-1
S50 Wolbachia endosymbiont of Drosophila melanogaster
S51 Silicibacter pomeroyi DSS-3
S52 Agrobacterium tumefaciens str. C58
S53 Agrobacterium tumefaciens str. C58
S54 Caulobacter crescentus CB15
S55 Brucella suis 1330
S56 Ehrlichia ruminantium str. Welgevonden
S57 Brucella abortus biovar 1 str. 9-941
S58 Rickettsia typhi str. Wilmington
S59 Zymomonas mobilis subsp. mobilis ZM4
S60 Wolbachia endosymbiont strain TRS of Brugia malayi
S61 Gluconobacter oxydans 621H
S62 Ehrlichia ruminantium str. Welgevonden
S63 Ehrlichia ruminantium str. Gardel
S64 Aquifex aeolicus VF5
S65 Bacteroides fragilis NCTC 9343
S66 Porphyromonas gingivalis W83
S67 Chlorobium tepidum TLS
S68 Bacteroides thetaiotaomicron VPI-5482
S69 Bacteroides fragilis YCH46
S70 Ralstonia solanacearum GMI1000
S71 Neisseria gonorrhoeae FA 1090
S72 Bordetella bronchiseptica RB50
S73 Bordetella parapertussis 12822
S74 Bordetella pertussis Tohama I
S75 Nitrosomonas europaea ATCC 19718
S76 Burkholderia mallei ATCC 23344
S77 Burkholderia pseudomallei K96243
S78 Neisseria meningitidis MC58
S79 Neisseria meningitidis Z2491
S80 Chromobacterium violaceum ATCC 12472
S81 Azoarcus sp. EbN1
S82 Chlamydia trachomatis D/UW-3/CX
S83 Chlamydophila caviae GPIC
S84 Chlamydia muridarum Nigg
S85 Chlamydophila pneumoniae AR39
S86 Chlamydophila pneumoniae CWL029
S87 Chlamydophila pneumoniae J138
S88 Chlamydophila abortus S26/3
S89 Chlamydophila pneumoniae TW-183
S90 Parachlamydia sp. UWE25
S91 Dehalococcoides ethenogenes 195
S92 Synechocystis sp. PCC 6803
S93 Prochlorococcus marinus subsp. pastoris str. CCMP1986
S94 Prochlorococcus marinus str. MIT 9313
S95 Synechococcus sp. WH 8102
S96 Nostoc sp. PCC 7120
S97 Thermosynechococcus elongatus BP-1
S98 Prochlorococcus marinus subsp. marinus str. CCMP1375
S99 Gloeobacter violaceus PCC 7421
S100 Synechococcus elongatus PCC 6301
S101 Deinococcus radiodurans R1
S102 Thermus thermophilus HB27
S103 Thermus thermophilus HB8
S104 Desulfovibrio vulgaris subsp. vulgaris str. Hildenborough
S105 Geobacter sulfurreducens PCA
S106 Bdellovibrio bacteriovorus HD100
S107 Desulfotalea psychrophila LSv54
S108 Campylobacter jejuni subsp. jejuni NCTC 11168
S109 Helicobacter hepaticus ATCC 51449
S110 Helicobacter pylori 26695
S111 Helicobacter pylori J99
S112 Campylobacter jejuni RM1221
S113 Wolinella succinogenes DSM 1740
S114 Staphylococcus epidermidis RP62A
S115 Enterococcus faecalis V583
S116 Lactococcus lactis subsp. lactis Il1403
S117 Bacillus cereus ATCC 10987
S118 Bacillus subtilis subsp. subtilis str. 168
S119 Clostridium acetobutylicum ATCC 824
S120 Clostridium perfringens str. 13
S121 Clostridium tetani E88
S122 Lactobacillus acidophilus NCFM
S123 Listeria monocytogenes str. 4b F2365
S124 Listeria innocua Clip11262
S125 Mycoplasma genitalium G-37
S126 Mycoplasma pneumoniae M129
S127 Mycoplasma pulmonis UAB CTIP
S128 Ureaplasma parvum serovar 3 str. ATCC 700970
S129 Mycoplasma penetrans HF-2
S130 Bacillus halodurans C-125
S131 Staphylococcus aureus subsp. aureus COL
S132 Thermoanaerobacter tengcongensis MB4
S133 Staphylococcus aureus subsp. aureus Mu50
S134 Staphylococcus aureus subsp. aureus N315
S135 Staphylococcus aureus subsp. aureus MRSA252
S136 Staphylococcus aureus subsp. aureus MSSA476
S137 Streptococcus pyogenes M1 GAS
S138 Listeria monocytogenes EGD-e
S139 Streptococcus pneumoniae TIGR4
S140 Streptococcus pneumoniae R6
S141 Staphylococcus epidermidis ATCC 12228
S142 Oceanobacillus iheyensis HTE831
S143 Streptococcus pyogenes MGAS8232
S144 Streptococcus pyogenes SSI-1
S145 Staphylococcus aureus subsp. aureus MW2
S146 Bacillus anthracis str. Ames
S147 Streptococcus pyogenes MGAS315
S148 Streptococcus agalactiae 2603V/R
S149 Streptococcus mutans UA159
S150 Streptococcus agalactiae NEM316
S151 Lactobacillus plantarum WCFS1
S152 Bacillus cereus ATCC 14579
S153 Mycoplasma gallisepticum R
S154 Onion yellows phytoplasma OY-M
S155 Lactobacillus johnsonii NCC 533
S156 Mycoplasma mycoides subsp. mycoides SC str. PG1
S157 Mesoplasma florum L1
S158 Mycoplasma mobile 163K
S159 Bacillus anthracis str. 'Ames Ancestor'
S160 Bacillus thuringiensis serovar konkukian str. 97-27
S161 Bacillus anthracis str. Sterne
S162 Bacillus licheniformis ATCC 14580
S163 Bacillus cereus E33L
S164 Streptococcus pyogenes MGAS10394
S165 Bacillus licheniformis ATCC 14580
S166 Mycoplasma hyopneumoniae 232
S167 Streptococcus thermophilus LMG 18311
S168 Streptococcus thermophilus CNRZ1066
S169 Geobacillus kaustophilus HTA426
S170 Bacillus clausii KSM-K16
S171 Fusobacterium nucleatum subsp. nucleatum ATCC 25586
S172 Francisella tularensis subsp. tularensis SCHU S4
S173 Xanthomonas campestris pv. campestris str. 8004
S174 Methylococcus capsulatus str. Bath
S175 Legionella pneumophila subsp. pneumophila str. Philadelphia
S176 Yersinia pestis CO92
S177 Vibrio cholerae O1 biovar eltor str. N16961
S178 Haemophilus ducreyi 35000HP
S179 Pasteurella multocida subsp. multocida str. Pm70
S180 Coxiella burnetii RSA 493
S181 Haemophilus influenzae Rd KW20
S182 Escherichia coli K12
S183 Escherichia coli O157:H7
S184 Salmonella enterica subsp. enterica serovar Typhi str. CT18
S185 Salmonella typhimurium LT2
S186 Buchnera aphidicola str. APS (Acyrthosiphon pisum)
S187 Buchnera aphidicola str. Bp (Baizongia pistaciae)
S188 Escherichia coli O157:H7 EDL933
S189 Pseudomonas putida KT2440
S190 Xylella fastidiosa 9a5c
S191 Wigglesworthia glossinidia endosymbiont of Glossina brevipa
S192 Xylella fastidiosa Temecula1
S193 Yersinia pestis KIM
S194 Xanthomonas campestris pv. campestris str. ATCC 33913
S195 Xanthomonas axonopodis pv. citri str. 306
S196 Shigella flexneri 2a str. 301
S197 Buchnera aphidicola str. Sg (Schizaphis graminum)
S198 Escherichia coli CFT073
S199 Pseudomonas syringae pv. syringae B728a
S200 Pseudomonas aeruginosa PAO1
S201 Shewanella oneidensis MR-1
S202 Vibrio vulnificus CMCP6
S203 Erwinia carotovora subsp. atroseptica SCRI1043
S204 Pseudomonas syringae pv. tomato str. DC3000
S205 Vibrio parahaemolyticus RIMD 2210633
S206 Salmonella enterica subsp. enterica serovar Typhi Ty2
S207 Shigella flexneri 2a str. 2457T
S208 Candidatus Blochmannia floridanus
S209 Vibrio vulnificus YJ016
S210 Photorhabdus luminescens subsp. laumondii TTO1
S211 Salmonella enterica subsp. enterica serovar Choleraesuis st
S212 Yersinia pestis biovar Medievalis str. 91001
S213 Idiomarina loihiensis L2TR
S214 Acinetobacter sp. ADP1
S215 Xanthomonas oryzae pv. oryzae KACC10331
S216 Yersinia pseudotuberculosis IP 32953
S217 Vibrio fischeri ES114
S218 Mannheimia succiniciproducens MBEL55E
S219 Salmonella enterica subsp. enterica serovar Paratyphi A str
S220 Legionella pneumophila str. Lens
S221 Legionella pneumophila str. Paris
S222 Photobacterium profundum SS9
S223 Rhodopirellula baltica SH 1
S224 Borrelia burgdorferi B31
S225 Treponema denticola ATCC 35405
S226 Treponema pallidum subsp. pallidum str. Nichols
S227 Leptospira interrogans serovar Lai str. 56601
S228 Leptospira interrogans serovar Copenhageni str. Fiocruz L1-
S229 Borrelia garinii PBi
S230 Thermotoga maritima MSB8

Figure 6. The Mapping between the Species and the Symbols.