Microbial Phylogenetic Context Using Phylogenetic Outlines

Caner Bagci[1], David Bryant[2], Banu Cetinkaya[3], and Daniel H. Huson[1,4,*]

[1]Algorithms in Bioinformatics, University of Tübingen, Germany

[2]Department of Mathematics, University of Otago, Dunedin, New Zealand

[3]Computer Science Program, Sabanci University, Tuzla/İstanbul, Turkey

[4]Cluster of Excellence: Controlling Microbes to Fight Infection, University of Tübingen, Tübingen, Germany

*Corresponding author: E-mail: daniel.huson@uni-tuebingen.de.

Accepted: 6 September 2021

Abstract

Microbial studies typically involve the sequencing and assembly of draft genomes for individual microbes or whole microbiomes. Given a draft genome, one first task is to determine its phylogenetic context, that is, to place it relative to the set of related reference genomes. We provide a new interactive graphical tool that addresses this task using Mash sketches to compare against all bacterial and archaeal representative genomes in the Genome Taxonomy Database taxonomy, all within the framework of SplitsTree5. The phylogenetic context of the query sequences is then displayed as a phylogenetic outline, a new type of phylogenetic network that is more general than a phylogenetic tree, but significantly less complex than other types of phylogenetic networks. We propose to use such networks, rather than trees, to represent phylogenetic context, because they can express uncertainty in the placement of taxa, whereas a tree must always commit to a specific branching pattern. We illustrate the new method using a number of draft genomes of different assembly quality.

Key words: phylogeny, genomes, *k*-mers, phylogenetic networks, algorithms, software.

Significance

Metagenomic sequencing allows the construction of draft genomes of unknown microbial species. There is a need for tools that make it easy to determine the possible taxonomic identity of such a genome. Here, we provide a fast and interactive software for computing the "taxonomic context" of a draft genome that is represented as a novel "phylogenetic outline."

Introduction

In the study of microbes using sequencing, assembly, and contig binning, one important task is to calculate the "phylogenetic context" of a given draft genome, contig, or bin of contigs. This requires that we first determine which known microbes have similar sequences to the query, and then produce a suitable indication of the phylogenetic relationships.

Pairwise distances between genome-scale sequences can be quickly calculated using *k*-mer methods such as Mash (Ondov et al. 2016). In this type of approach, the *k*-mer content (words of a fixed length *k*) of a sequence is represented by a reduced "sketch" and such sketches are compared using the Jaccard index and derived distance measures that approximate average nucleotide identity (ANI).

The Genome Taxonomy Database (GTDB) (Parks et al. 2020) provides a similarity-based taxonomy for ≈195,000 bacterial and archaeal genomes obtained from the NCBI assembly database (Kitts et al. 2016). A representative subset of ≈32,000 reference genomes is provided for taxonomic analysis and the GTDB-tk tool kit provides associated analysis tools (Chaumeil et al. 2019).

Here, we propose to compute a Mash sketch for each representative reference genome in the GTDB, and to assign a Bloom filter (Bloom 1970) to each internal node of the taxonomy so as to represent the set of all *k*-mers present in reference genomes below the node (Solomon and Kingsford 2016; Pierce et al. 2019). For a given set of query sequences, this will allow one to determine all similar reference genomes quickly enough for use in an interactive program. Mash can then be used to compute a distance matrix on the query and (a subset of) all sufficiently similar references.

Given such a matrix of pairwise distances, one option is to compute a phylogenetic tree to represent the data, using an algorithm such as neighbor-joining (Saitou and Nei 1987). Phylogenetic trees are often used to represent such data, because evolution is assumed to be predominantly driven by speciation events. In addition, phylogenetic trees have low complexity, employing only a linear number $O(n)$ of nodes and edges to represent n taxa.

However, in the evolution of microbes, reticulate events, such as horizontal gene transfer and recombination, may play a significant role (Huson et al. 2010). In addition, when using *k*-mer features and distance-based phylogenetic methods, the accuracy of the resulting phylogenetic trees may be poor. Hence, the use of phylogenetic networks, rather than phylogenetic trees, can be more appropriate.

One popular approach to obtaining a phylogenetic network (Huson and Bryant 2006) is to apply the neighbor-net algorithm (Bryant and Moulton 2004) on the distances and to represent the output as a splits network (Dress and Huson 2004), requiring $O(n^4)$ nodes and edges, in the worst case.

Here, we present a new type of phylogenetic network that we call a "phylogenetic outline" (fig. 1). A phylogenetic outline is also computed from the output of the neighbor-net algorithm and has the mathematical properties of a splits network. It displays all the calculated splits, but uses substantially fewer nodes and edges to do so. Indeed, phylogenetic outlines are only quadratic in size, containing at most $O(n^2)$ nodes and edges. By default, phylogenetic outlines are unrooted, however, we also provide algorithms for both midpoint and outgroup rooting.

Although our focus here is on using phylogenetic outlines to represent phylogenetic context, please note that phylogenetic outlines can be used to represent the output of the neighbor-net algorithm in all other settings, as well.

The entire procedure described here has been implemented as part of SplitsTree5. The implementation carries out a Mash comparison of a set of query sequences against a database representing the GTDB, so as to determine the phylogenetic context of the queries, then computes and visualizes a phylogenetic outline of the sequences.

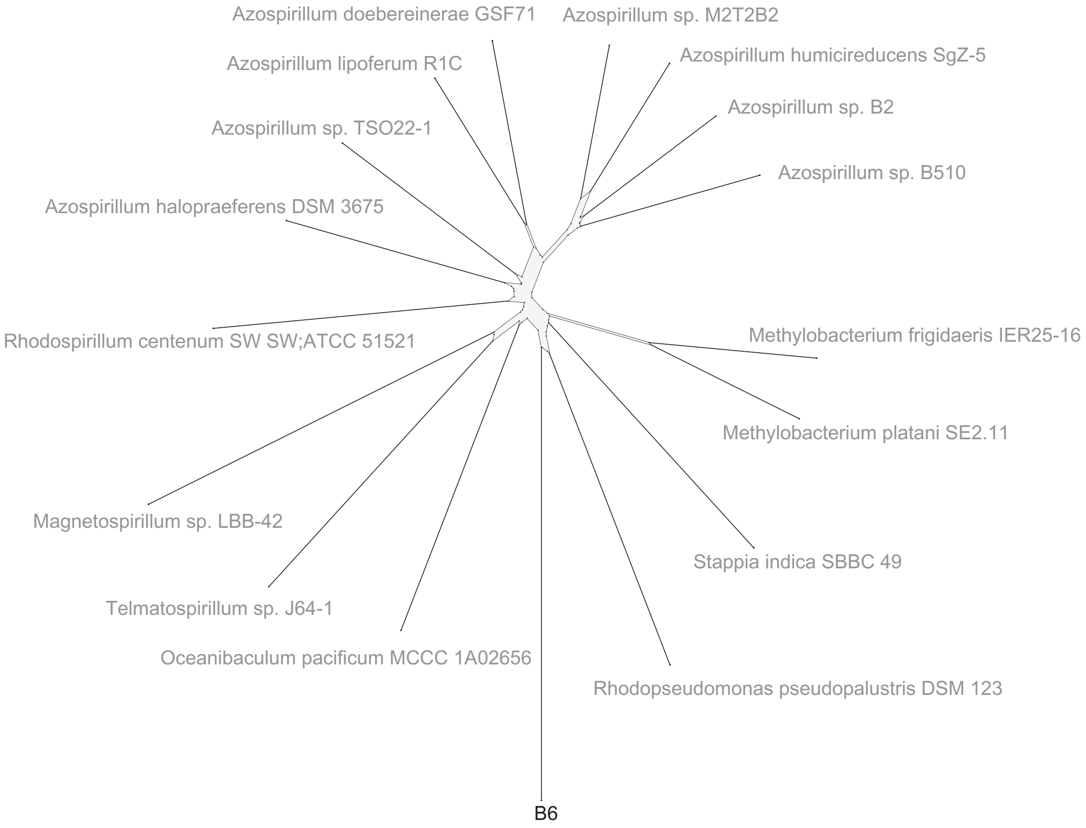

Azospirillum doebereinerae GSF71
Azospirillum sp. M2T2B2
Azospirillum lipoferum R1C
Azospirillum humicireducens SgZ-5
Azospirillum sp. TSO22-1
Azospirillum sp. B2
Azospirillum sp. B510
Azospirillum halopraeferens DSM 3675
Rhodospirillum centenum SW SW;ATCC 51521
Methylobacterium frigidaeris IER25-16
Methylobacterium platani SE2.11
Magnetospirillum sp. LBB-42
Stappia indica SBBC 49
Telmatospirillum sp. J64-1
Oceanibaculum pacificum MCCC 1A02656
Rhodopseudomonas pseudopalustris DSM 123
B6

Fig. 1.—A phylogenetic outline, displaying the phylogenetic context of the metagenomic draft genome B6 from Arumugam et al. (2019).

Using a single dialog, the user selects the files containing the query sequences, loads a database containing all reference data and then obtains a phylogenetic outline of the queries, interactively in minutes. Unlike other approaches (Ondov et al. 2016; Chaumeil et al. 2019; Pierce et al. 2019), no scripting or running of multiple programs is required.

Conceptually, the calculation of "phylogenetic context" lies between "phylogenetic placement" (Matsen et al. 2010), in which one or more query sequences are placed into a precomputed phylogenetic tree, and ab initio phylogenetic tree inference, in which a phylogenetic tree is calculated for all query sequences and a subset of the reference sequences. The GTDB-tk toolkit provides tools for performing phylogenetic placement and ab initio tree inference. In both cases, the result is a phylogenetic tree that can be viewed in a program such as Dendroscope (Huson and Scornavacca 2012).

To illustrate our method, we apply it to a number of metagenomic draft genomes of different levels of quality, published in Arumugam et al. (2019). We also show how this differs from the phylogenetic analyses that one can perform using GTDB-tk.

Results

Assume that you have sequenced and assembled one or more bacterial genomes, or have calculated a metagenomic binning of contigs. There are a number of command-line pipelines that can be used to determine closely related genomes, ranging from very fast, *k*-mer based heuristics such as Mash (Ondov et al. 2016), Sourmash (Pierce et al. 2019), or marker-gene based phylogenetic placement methods such as GTDB-tk (Parks et al. 2020), to more thorough, but slower protein-alignment based approaches such as DIAMOND+MEGAN (Buchfink et al. 2015; Huson et al. 2016) or HUMAnN2 (Franzosa et al. 2018). These methods all require scripting to go from an input file containing one or more sequences of interest to a visualization of the phylogenetic context of the input sequences. Moreover, the visual representation of the context is often performed using a phylogenetic or taxonomic tree, which presents a definite clustering of taxa with little indication of uncertainty or alternative groupings.

The shortcomings of using a single phylogenetic tree to represent uncertain data are well known and have been addressed in number of different approaches, such as consensus networks (Holland et al. 2004), DensiTree visualizations (Bouckaert 2010), or the "branch parsimony score" that aims at quantifying uncertainty in sample placements (Turakhia et al. 2021), to name a few.

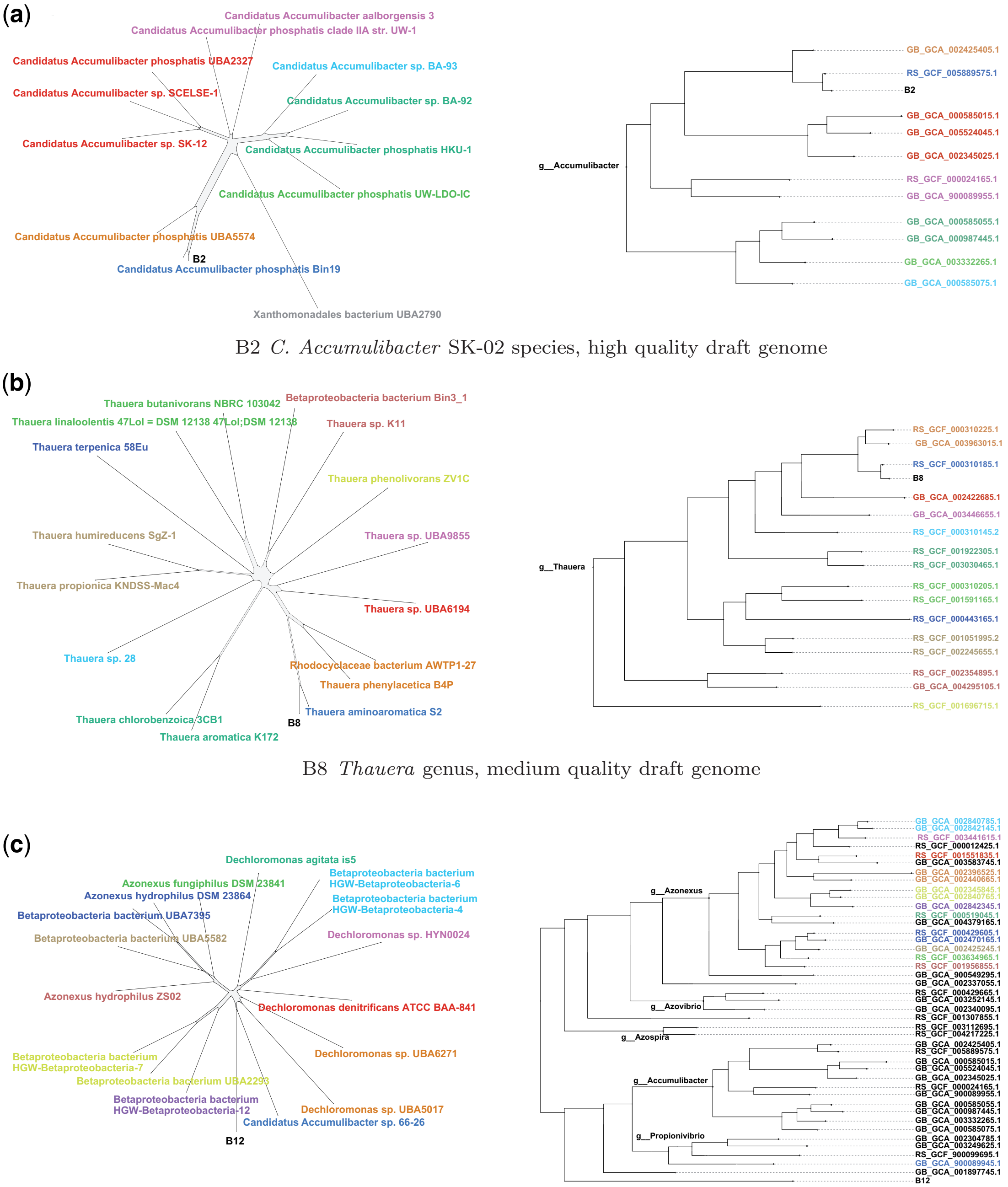
We provide a fast and interactive implementation for exploring phylogenetic context of a set of microbial sequences of interest. The user loads one or more files of query DNA sequences and then requests that all similar reference genomes are determined. Then a threshold is set for the maximum distance of reference genomes, or number of reference genomes, to be considered. These are downloaded and a Mash comparison of the query sequences and all similar reference genomes is performed, the neighbor-net method is run, and the result is presented as a phylogenetic outline. (The user can also choose to use a tree-building method such as neighbor-joining; Saitou and Nei 1987).

To illustrate our method, we applied it to a number of "draft genomes" reported in Arumugam et al. (2019). These draft genomes contain assembled contigs of long-read microbiome sequences obtained from a bio-reactor enriched for polyphosphate accumulation (Each such draft genome is a "metagenomic assembly bin" that consists of one or more contigs that are deemed to belong to the same genome.). The paper reports a taxonomic assignment for each bin that is based on an analysis of the contained protein-coding genes and confirmed using 16S rRNA sequences, when present. For each of the 14 reported draft genomes, we calculated a phylogenetic outline to display the phylogenetic context of the closest reference genomes below a certain distance.

On the left of figure 2, we show one "high-quality" draft genome (that is, with $> 90\%$ completeness and $< 5\%$ contamination), one "medium-quality" draft genome (with $\geq 50\%$ completeness and $< 10\%$ contamination), and one "low-quality" draft genome (with $< 50\%$ completeness and $< 10\%$ contamination), respectively. See Bowers et al. (2017) for the definition of the three quality levels in terms of completeness and contamination. The other 11 bins are shown in the Supplementary Material online.

Generally speaking, in all three cases, the phylogenetic context is compatible with the taxonomic assignment reported in Arumugam et al. (2019). In the case of draft genome B2, all (but one) reference genomes displayed in the phylogenetic context are members of the genus *Candidatus Accumulibacter*. This is in agreement with the classification presented in Arumugam et al. (2019), which assigned B2 to the species *Candidatus Accumulibacter* sp. SK-02. The closest species in the phylogenetic context analysis is *Candidatus Accumulibacter phosphatis* Bin19, Mash distance 0.01, a genome that was not available to Arumugam et al. (2019). The second closest, *Candidatus Accumulibacter phosphatis* UBA5574, Mash distance 0.07, is not represented by protein sequences in NCBI and thus was not part of the database used by Arumugam et al. (2019).

Unexpectedly, the species *Xanthomonadales bacterium* UBA2790, which comes from a different taxonomic class, also appears in the phylogenetic context of B2, with a Mash distance of 0.2. Note that this metagenome-assembled genome (MAG) comes from a sample of granular sludge that also gave rise to two *Candidatus Accumulibacter* reference genomes (Parks et al. 2017) and we suspect that it might


(a)
Candidatus Accumulibacter aalborgensis 3
Candidatus Accumulibacter phosphatis clade IIA str. UW-1
Candidatus Accumulibacter phosphatis UBA2327
Candidatus Accumulibacter sp. BA-93
Candidatus Accumulibacter sp. SCELSE-1
Candidatus Accumulibacter sp. BA-92
Candidatus Accumulibacter sp. SK-12
Candidatus Accumulibacter phosphatis HKU-1
Candidatus Accumulibacter phosphatis UW-LDO-IC
Candidatus Accumulibacter phosphatis UBA5574
B2
Candidatus Accumulibacter phosphatis Bin19
Xanthomonadales bacterium UBA2790
g__Accumulibacter
GB_GCA_002425405.1
RS_GCF_005889575.1
B2
GB_GCA_000585015.1
GB_GCA_005524045.1
GB_GCA_002345025.1
RS_GCF_000024165.1
GB_GCA_900089955.1
GB_GCA_000585055.1
GB_GCA_000987445.1
GB_GCA_003332265.1
GB_GCA_000585075.1

B2 *C. Accumulibacter* SK-02 species, high quality draft genome

B8 *Thauera* genus, medium quality draft genome

B12 Betaproteobacteria class, low quality draft genome

Fig. 2.—Phylogenetic context and placement. For three metagenomic draft genomes B2, B8, and B12, we report the taxonomic assignment and genome quality (Arumugam et al. 2019), and display both the phylogenetic outline computed by SplitsTree5 and a tree representing the phylogenetic placement computed using GTDB-Tk.

be contaminated with *Candidatus Accumulibacter* contigs or sequence.

In the case of draft genome B8, all references genomes displayed in the phylogenetic context are *Thauera* species, except one unclassified Betaproteobacteria bacterium, which is, however, a member of the genus *Thauera*, and this supports the assignment to the genus *Thauera*. The closest reference genome *Thauera aminoaromatica* S2 has a Mash distance of 0.2.

Finally, in Arumugam et al. (2019), the draft genome B12 was classified as a member of the Betaproteobacteria class, suggesting that there did not exist a closely related reference at the time of the publication of the data set. In the phylogenetic context computed by SplitsTree, B12 is placed closest to

Candidatus Accumulibacter sp. 66-26 with a Mash distance of 0.14. In addition, some species of the genera *Azonexus and Dechloromonas* can be found that are similar to B12, with Mash distances below 0.18. These two genera belong to the family of Azonexaceae in the NCBI taxonomy, whereas *Candidatus Accumulibacter* does not have a defined family or order. All three genera belong to the family Rhodocyclaceae in the GTDB taxonomy. Although B12 does not have many closely related reference genomes, the phylogenetic outline produced by SplitsTree5 suggests that B12 belongs to the Rhodocyclaceae family, which is more specific that the assignment suggested in Arumugam et al. (2019).

In each of the three examples, it took between 1 and 3 min to determine all reference genomes whose sketches have a distance of at most 0.3 to the sketch of the draft genome, and then to compute and display the phylogenetic outlines for the ten most similar references. Computations were carried out on a laptop with eight cores (at 2.4 GHz) and 32 GB of memory. Reference genomes are downloaded (and cached) on demand, which takes additional time. The distance thresholds used to select the closest reference genomes for each bin were chosen interactively and are reported in the Supplementary Material online.

To illustrate the improved practical performance of the outline algorithm on a larger data set, we computed the phylogenetic context for draft genome B12 using the 1,000 closest reference genomes. Running the neighbor-net algorithm on this data takes 90 s and results in 4,516 splits. The equal-angle algorithm (Dress and Huson 2004) produces a splits network with 108,640 nodes and 212,762 edges, and requires about 7 min to compute and show the network. In contrast, our new outline algorithm produces a splits network with 8,028 nodes and 8,028 edges and requires only 2 s for this (not shown here).

For the purpose of comparison, we applied GTDB-Tk (Chaumeil et al. 2019) in phylogenetic placement mode (classify_wf workflow) to the draft genomes B2, B8, and B12. GTDB-Tk uses GTDB accessions to label reference genomes, whereas SplitsTree uses the strain names associated with the assemblies in the assembly reports of NCBI. In figure 2, we show relevant part of the placement tree computed by GTDB-tk and use colors to indicate corresponding GTDB accessions and NCBI strain names.

In the case of draft genome B2, the tree computed by GTDB-Tk agrees very well with the phylogenetic context computed using SplitsTree, placing B2 next to *Candidatus Accumulibacter phosphatis* Bin19, and to other reference genomes shown in the phylogenetic outline (fig. 2*a*). The distances computed by SplitsTree5 were also similar to those reported by GTDB-Tk.

In the case of draft genome B8, the phylogenetic context included all members of the genus *Thauera* from GTDB-Tk, and placed the query next to *Thauera aminoaromatica* S2. The tree produced by GTDB-Tk contains the same references and has a similar topology (fig. 2*b*). This suggests that, if distances between genomes are small enough, then a Mash-based analysis, as in SplitsTree5, may perform very similar to a marker-gene and ANI-based analysis, as in GTDB-Tk.

Finally, in the case of B12, this draft genome is further away from any reference genome than the two draft genomes just discussed (fig. 2*c*). GTDB-Tk places B12 outside of the boundaries of any genera, but closer to the genus *Accumulibacter*, and closest to the species *Candidatus Accumulibacter* sp. 66-26.

SplitsTree5 also places B12 closest to the species *Candidatus Accumulibacter* sp. 66-26; however, the rest of the references shown in the phylogenetic context are from the genus *Azonexus* instead of *Accumulibacter*. Although the distances within the genus are in agreement with those computed by GTDB, here, we see a difference in the phylogeny outside genus boundaries. This is reflects the fact that ANI values are known to provide only a poor estimation of evolutionary distance across different genera (Qin et al. 2014).

We also determined the phylogenetic context and GTDB-Tk placement for all other 11 MAGs reported in Arumugam et al. (2019) and report these in the Supplementary Material online. For those draft genomes for which very similar reference genomes can be found, the phylogenetic context computed by SplitsTree is similar to the phylogenetic placement computed by GTDB-Tk. In the other cases, either the phylogenetic context contains only very few references, or it contains a wide range of different references and disagrees with the phylogenetic placement computed by GTDB-Tk (see B4 and B6 in the Supplementary Material online). These disagreements persist even if one uses a more accurate calculation of ANI (not shown here), indicating that they are due to a fundamental difference between ANI analysis and marker-gene analysis.

Discussion

Here, we bring together a number of different ideas, using the GTDB database to represent the taxonomy of bacterial and archaeal genomes; Mash sketches and Bloom filters for fast sequence comparison; and the neighbor-net method and our new concept of phylogenetic outlines for visualization. We thus provide a fast heuristic for establishing the phylogenetic context for one or more prokaryotic genomes or DNA sequences. We demonstrated that our approach can be applied to usefully determine and visualize the phylogenetic context of bacterial draft genomes at different levels of assembly quality.

We believe that the use of a phylogenetic outline, rather than a phylogenetic tree, to represent phylogenetic context is more suitable because outlines can express vagueness in the placement of taxa with respect to each other, whereas trees suggest a specific branching pattern. For example, in figure 4*a*, we show the unrooted, resolved phylogenetic tree computed using the neighbor-joining algorithm (Saitou and Nei 1987). Both the splits network (fig. 4*b*) and the phylogenetic outline (fig. 4*c*) place *Competibacteraceae bacterium UBA2788* halfway between *Candidatus Contendobacter odensis Run B J11* and the draft genome B11. This ambiguity of placement is not evident in the tree representation.

The mash-based calculation of phylogenetic outlines presented here is, on the one hand, a form of ab-initio phylogenetic analysis, in which we infer evolutionary relationships from data. On the other hand, the aim is to visualize the phylogenetic context of sequences, which is similar to the goal of phylogenetic placement. We provide a graphical user interface that allows the user to interactively compute and explore the context of sequences. Although GTDB-tk provides methods for both phylogenetic placement and ab initio phylogenetic analysis, these calculations are performed using a script and the output is presented as text files. Although the resulting trees can be viewed using third party tools, the leaves of the tree are labeled by GTDB accessions, whereas our approach provides the option of labeling leaves by the associated NCBI names.

Here, the focus is on prokaryotic sequences. It would be straight-forward to adopt this approach to eukaryotes with small genomes, such as *Phytophthora* or certain insects, say. Virus genomes such as HIV or SARS-2-COVID, are too small to benefit from a naïve sketching approach, whereas mammalian genomes are probably too big to handle with our current code base. Using mash to screen sequencing data for the presence of certain genomes is an attractive idea (Ondov et al. 2019), but it exceeds the envisioned scope of this software.

Phylogenetic outlines are not limited to depicting phylogenetic context and we envision them becoming the preferred visualization of the output of the neighbor-net algorithm in other types of analysis, too.

Using a phylogenetic outline to represent phylogenetic context does not replace careful alignment and sophisticated phylogenetic analysis when the goal is to understand the evolutionary history of a set of taxa in detail. In addition, contamination of metagenomic assembly bins may cause difficulties. Nevertheless, we believe that our approach will prove to be a useful addition to the biologists' computational toolbox.

Materials and Methods

Preprocessing the Reference Database

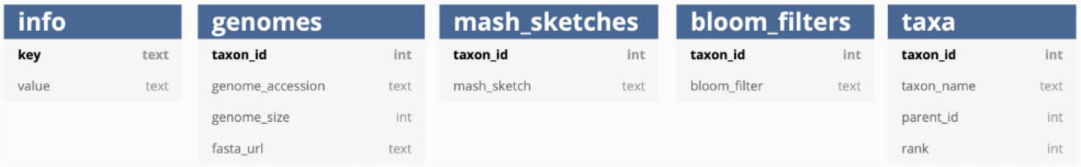
We downloaded the GTDB taxonomy (Parks et al. 2020) in July 2020. The taxonomy has 240,103 nodes, of which 194,600 are leaves. GTDB identifies 31,910 genomes representative genomes. These are available from the GTDB download page https://data.gtdb.ecogenomic.org/releases/latest/genomic_files_reps/ (last accessed September 16, 2021). Links to the other (nonrepresentative) genomes are contained in the GenBank or RefSeq (Pruitt et al. 2009) assembly summary reports on the NCBI genomes FTP site ftp://ftp.ncbi.nlm.nih.gov/genomes/ASSEMBLY_REPORTS/.

In a processing step, we computed a Mash sketch (Ondov et al. 2016) for each of the 31,910 representative genomes, using a word size of $k = 21$ and sketch size of $s = 10{,}000$. Multipart genome sequences were concatenated. For each internal node of the GTDB taxonomy, we computed a Bloom filter (Bloom 1970) representing all k-mers contained in all sketches associated with genomes below the node, using a false positive probability of 0.0001. For these calculations, we used our own implementations of the Mash algorithm, mash-sketches, and Bloom filters, bfilter-tool, which we provide as a part of our SplitsTree5 package.

All taxa, Mash sketches, Bloom filters, and genome URL's were loaded into an SQLITE database file gtdb-rep-k21-s10000-May2021.db. In addition, the file contains an explicit representation of the GTDB taxonomy using a node-to-parent mapping. The database schema is shown in figure 3. The database file is 12.4 GB in size and does not contain the actual genome sequences; these are downloaded (and cached) by our implementation on demand.

The Outline Algorithm

For a given distance matrix D on a set of n taxa $\mathcal{X}$, the neighbor-net algorithm (Bryant and Moulton 2004) computes a set of weighted splits Σ of $\mathcal{X}$, that is, a set of bipartitions of the form $S = A|B$, where $A \neq \emptyset$, $B \neq \emptyset$, $A \cap B = \emptyset$ and $A \cup B = \mathcal{X}$. The set of splits computed by neighbor-net has quadratic size $O(n^2)$. The set of splits is "circular," which implies that they can be represented by an "outer-labeled



info	
key	text
value	text

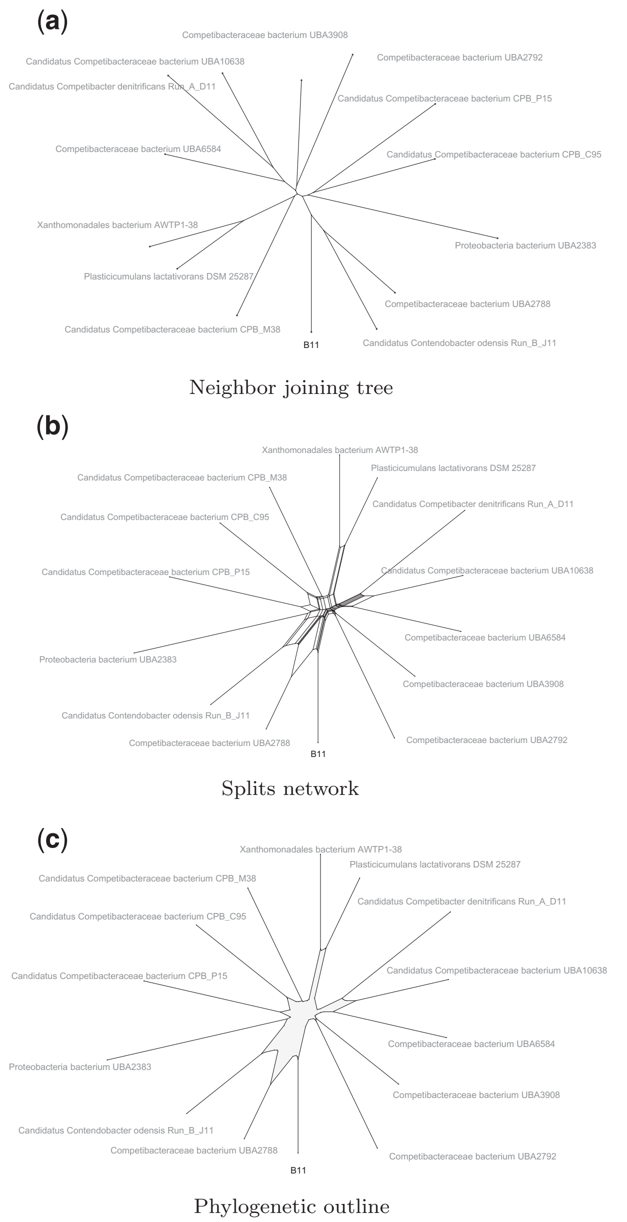
genomes	
taxon_id	int
genome_accession	text
genome_size	int
fasta_url	text

mash_sketches	
taxon_id	int
mash_sketch	text

bloom_filters	
taxon_id	int
bloom_filter	text

taxa	
taxon_id	int
taxon_name	text
parent_id	int
rank	int

Fig. 3.—Database schema. The info table contains general information, such as version and size of the database. The primary key for all other tables is the taxon ID. For each reference species, the genomes table contains the genome accession, genome size, and the URL of a FastA file containing the genome sequence. The mash_sketches table contains a mash sketch for each reference species, whereas the bloom_filters tables contains a Bloom filter for each higher-rank taxon. The taxa table contains the ID and name, the parent ID, and the rank, for each taxon.


(a)
Competibacteraceae bacterium UBA3908
Candidatus Competibacteraceae bacterium UBA10638
Competibacteraceae bacterium UBA2792
Candidatus Competibacter denitrificans Run_A_D11
Candidatus Competibacteraceae bacterium CPB_P15
Competibacteraceae bacterium UBA6584
Candidatus Competibacteraceae bacterium CPB_C95
Xanthomonadales bacterium AWTP1-38
Proteobacteria bacterium UBA2383
Plasticicumulans lactativorans DSM 25287
Competibacteraceae bacterium UBA2788
Candidatus Competibacteraceae bacterium CPB_M38
B11
Candidatus Contendobacter odensis Run_B_J11
(b)
Xanthomonadales bacterium AWTP1-38
Candidatus Competibacteraceae bacterium CPB_M38
Plasticicumulans lactativorans DSM 25287
Candidatus Competibacter denitrificans Run_A_D11
Candidatus Competibacteraceae bacterium CPB_C95
Candidatus Competibacteraceae bacterium CPB_P15
Candidatus Competibacteraceae bacterium UBA10638
Competibacteraceae bacterium UBA6584
Proteobacteria bacterium UBA2383
Competibacteraceae bacterium UBA3908
Candidatus Contendobacter odensis Run_B_J11
Competibacteraceae bacterium UBA2788
Competibacteraceae bacterium UBA2792
B11
(c)
Xanthomonadales bacterium AWTP1-38
Plasticicumulans lactativorans DSM 25287
Candidatus Competibacteraceae bacterium CPB_M38
Candidatus Competibacter denitrificans Run_A_D11
Candidatus Competibacteraceae bacterium CPB_C95
Candidatus Competibacteraceae bacterium CPB_P15
Candidatus Competibacteraceae bacterium UBA10638
Competibacteraceae bacterium UBA6584
Proteobacteria bacterium UBA2383
Competibacteraceae bacterium UBA3908
Candidatus Contendobacter odensis Run_B_J11
Competibacteraceae bacterium UBA2788
Competibacteraceae bacterium UBA2792
B11

Neighbor joining tree

Splits network

Phylogenetic outline

Fig. 4.—Tree and networks. For a low-quality draft genome “B11” from Arumugam et al. (2019), we display its calculated phylogenetic context, using (*a*) a neighbor-joining tree with 26 nodes and 25 edges, (*b*) a splits network with 120 nodes and 197 edges, and (*c*) a phylogenetic outline with 68 nodes and 68 edges, respectively.

planar” splits network (Dress and Huson 2004) (fig. 4*b*), using $O(n^4)$ nodes and edges, in the worst case.

Here, we describe the computation of a phylogenetic outline that requires only $O(n^2)$ nodes and edges (fig. 4*c*). In a phylogenetic outline, each split $S = A|B$ is represented by a single edge, or two parallel edges, that separate all taxa in A from all taxa in B, and thus a phylogenetic outline fulfills the definition of a splits network (Dress and Huson 2004).

Consider a set Σ of m splits on $\mathcal{X}$, each split S with a positive weight $\omega(S)$. Assume, without loss of generality, that Σ contains all trivial splits on $\mathcal{X}$, that is, all splits that separate exactly one taxon from all others. We will assume that the splits are circular, that is, that there exists an ordering $x_1, x_2, \ldots, x_n$ of the taxon set $\mathcal{X}$ such that each split $S \in \Sigma$ can be written as $S = \{x_i, \ldots, x_j\}|\mathcal{X} - \{x_i, \ldots, x_j\}$, with $1 < i \le j \le n$, in other words, as an interval of elements of $\mathcal{X}$, which does not contain the first taxon, versus all others. This condition is always satisfied by the output of neighbor-net (Bryant and Moulton 2002).

To illustrate this, consider the set of splits $\mathcal{S} = \{S_1, \ldots, S_5, S_a, S_b, S_c\}$ on $\mathcal{X} = \{x_1, \ldots, .x_5\}$, where $S_a = \{x_2, x_3\}|\{x_1, x_4, x_5\}$, $S_b = \{x_3, x_4, x_5\}|\{x_1, x_2\}$ and $S_c = \{x_3, x_4\}|\{x_1, x_2, x_5\}$. Moreover, for $i = 1, \ldots 5$, let S_i be the trivial split separating x_i from all other taxa. This set of splits is circular, as illustrated in figure 5*a*.

Circularity implies that, for each split $S \in \Sigma$, the split part not containing x_1 is an interval of the form $I(S) = \{x_i, \ldots, x_j\}$ with $1 < i \le j \le n$. We will use *i*(*S*) and *j*(*S*) to refer to the two interval bounds.

Our new “outline algorithm” for computing a phylogenetic outline proceeds in three steps. In summary, first, we define two “events” per split. Second, we sort all events. Third, we process all events in sorted order, constructing either 0 or 1 new nodes and/or edges, per event.

For each split S, we define two events, an “outbound event” S^+, crossing over to the other side of S from the side that contains x_1, and an “inbound event” S^-, returning back to the side of S that contains x_1. We will sort these events and then use them to construct the phylogenetic outline.

We define a total ordering on all events as follows (fig. 5*b and c*):

- For two outbound events S^+ and T^+, set $S^+ < T^+$, if either $i(S) < i(T)$ or both $i(S) = i(T)$ and $j(S) > j(T)$.
- For two inbound events S^- and T^-, set $S^- < T^-$, if either $j(S) < j(T)$ or both $j(S) = j(T)$ and $i(S) > i(T)$.
- For an outbound event S^+ and an inbound event T^-, set $S^+ < T^-$, if $i(S) < j(T) + 1$, and set $S^+ > T^-$, otherwise.

The ordering of all $O(n^2)$ events can be computed in $O(n^2)$ steps: Use radix sort to first sort all outbound events S^+ in decreasing order of *j*(*S*), and then in increasing order of *i*(*S*). Similarly, use radix sort to first sort all inbound events S^- in decreasing order of *i*(*S*) and then in increasing order of *j*(*S*). Finally merge the two lists of events observing the relative ordering of outbound and inbound events.

We now describe how to create the nodes and edges of the outline (fig. 5*d*).

We will use p to denote the current location, initially set to (0, 0). Place taxon x_1 on a new node v_1 at location $\pi(v_1) = p$. We will use $\Sigma(v)$ to denote the set of splits that separates a node v from the node v_1.

For each split S, we define an angle:

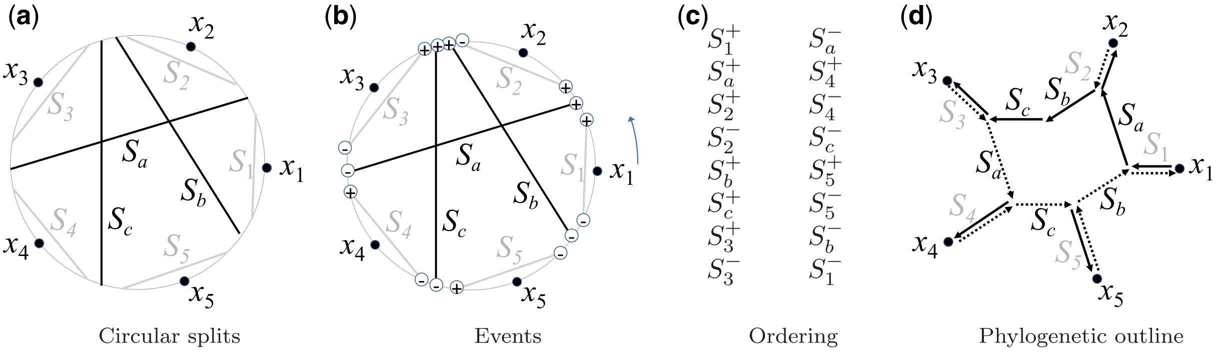

(a)
x_1
x_2
x_3
x_4
x_5
S_1 S_2 S_3 S_4 S_5 S_a S_b S_c
Circular splits
(b)
Events
(c)
S_1^+ S_a^+ S_2^+ S_2^- S_b^+ S_c^+ S_3^+ S_3^-
S_a^- S_4^+ S_4^- S_c^- S_5^+ S_5^- S_b^- S_1^-
Ordering
(d)
Phylogenetic outline

Fig. 5.—Circular splits and outline. (*a*) A set of splits $\{S_1,\ldots,S_5,S_a,S_b,S_c\}$ that is circular, that is, for which the taxa can be placed around a circle such all splits correspond to chords of the circle. (*b*) Traveling around the circle in positive orientation, each split *S* is encountered twice, first where the interval that does not contain taxon x_1 starts (outbound event S^+ marked ⊕) and then again where that interval ends (inbound event S^- marked ⊖). (*c*) The ordered events, listed in two columns. (*d*) Starting at x_1, in the order of the events, when encountering an outbound event S^+ move perpendicularly to the chord for split *S* by a distance of $\omega(S)$, as indicated by solid arrows. When encountering an inbound event S^-, move in the opposite direction by the same distance, as indicated by dotted arrows.

$$\alpha(S)=\frac{i(S)+j(S)-2}{2n}360^\circ.$$

In the example in figure 5*b*, this is perpendicular to the chord associated with *S*.

We process all events as described in the following two paragraphs. Let *v* be the current node, initially set to v_1.

To process an outbound event for a split *S*, move the current location *p* in the direction of $\alpha(S)$ by a distance of $\omega(S)$, the given positive weight of *S*. Create a new node *w* and connect *v* to *w* by a new edge. Set $\Sigma(w)=\Sigma(v)\cup\{S\}$. Update the current node, setting $v=w$.

To process an inbound event for a split *S*, move the current location *p* in the opposite direction of $\alpha(S)$ by a distance of $\omega(S)$. Consider the set of splits $\Sigma'=\Sigma(v)-\{S\}$. We set $w=u$, if there exists a node *u* with $\Sigma(u)=\Sigma'$. Else, we create a new node *w*, and set $\pi(w)=p$ and $\Sigma(w)=\Sigma'$. We connect *v and w* by an edge, if they are not already connected by an edge. Update the current node, setting $v=w$.

After processing all events, we arrive back at the starting point (0, 0); this is due to the fact that translations are commutative and so, for each split, the effect of processing its outbound event and the effect of later processing its inbound event cancel each other out.

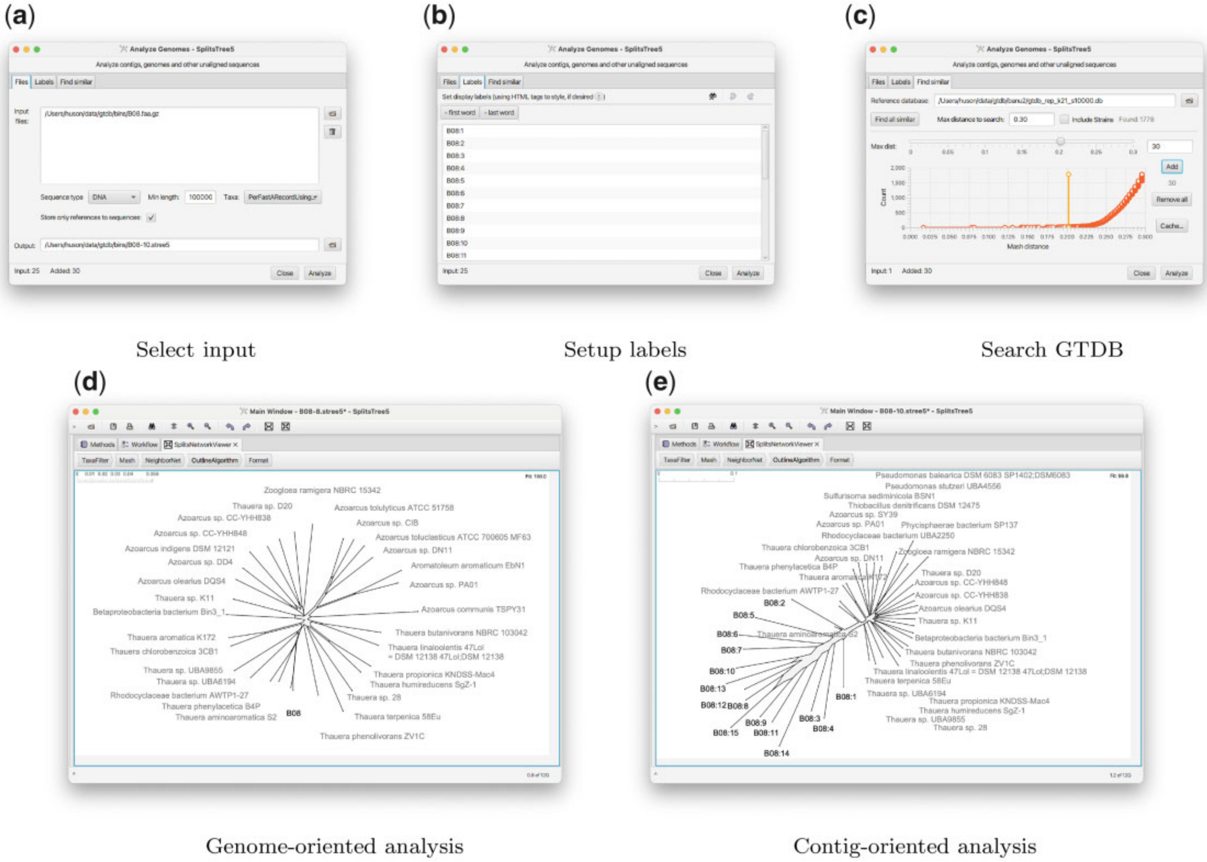
The number *m* of circular splits on *n* taxa is bounded by $O(n^2)$. As discussed above, there will be at most 2*m* nodes and 2*m* edges in the network, and therefore the size is bounded by $O(n^2)$. The events are sorted using radix sort, in time linear in the number of events, and thus in $O(n^2)$ time. The construction of nodes and edges also requires only $O(n^2)$ steps. Hence, the outline algorithm requires at most $O(n^2)$ in total. The network size and time requirement compare favorably to the $O(n^4)$ network size and time worst-case requirements of the equal angle algorithm (Dress and Huson 2004), which is currently used to visualize the output of the neighbor-net algorithm in SplitsTree4 (Huson and Bryant 2006).

We now discuss how to compute a rooted phylogenetic outline. For midpoint rooting, we proceed as follows. We first determine two taxa, *a and b*, that maximize the split distance $d_\Sigma(a,b)=\sum_{S\in\Sigma(a,b)}\omega(S)$, where the sum is taken over the set $\Sigma(a,b)$ of all splits *S* that separate *a and b*. The set $\Sigma(a,b)$ is then sorted by increasing cardinality of the split part *S*(*a*) containing *a*, and then by increasing size of the intersection of *S*(*a*) with the interval of all taxa that lie between *a and b* in the cycle. The root is then positioned in the first split for which the accumulated sum of weights is at least half of $d_\Sigma(a,b)$. For rooting by outgroup, the root is placed in the middle of a split that separates the outgroup from the rest of the taxa and is minimal with respect to that property.

Graphical User Interface

We have implemented the approach described here in our program SplitsTree5. To compute a phylogenetic outline displaying the phylogenetic context for one or more prokaryotic sequences, select the File → Analyze Genomes... menu item. This will open a dialog with three tabs. The first tab is used to select the input file(s) and output file, and to determine whether all sequences in a given file are to be concatenated or to be treated separately (fig. 6*a*). In addition, one can set a minimum sequence length (here set to 100,000 bp). The second tab is used to edit the names for the sequences (fig. 6*b*). The third tab is used to perform a Mash-based search in the GTDB database and to select which reference genomes should be included in the phylogenetic outline, based on their distances to the input sequences (fig. 6*c*).

The example presented is a medium-quality draft genome consisting of 25 contigs assembled from long-read sequences, designated bin B8 in Arumugam et al. (2019) with taxonomic


(a)
(b)
(c)
(d)
(e)
Analyze Genomes - SplitsTree5
Main Window - B08-8.stree5* - SplitsTree5
Main Window - B08-10.stree5* - SplitsTree5

Select input

Setup labels

Search GTDB

Genome-oriented analysis

Contig-oriented analysis

Fig. 6.—Phylogenetic context analysis. (*a*) The user selects the input file(s) and decides whether to analyze on a "per file" (complete genome) or "per FastA record" (individual contigs) basis. (*b*) The labels are set for the input sequences. (c) A search against the GTDB database is initiated and a threshold for the maximum distance is set. Once completed, a phylogenetic outline is drawn. (*d*) In a genome-oriented analysis, the phylogenetic outline shows the context of the concatenated input sequences. (*e*) Alternatively, in a contig-oriented analysis, the different sequences in the input file are represented individually in the phylogenetic outline.

assignment to the genus *Thauera*. In figure 6*d*, we use a phylogenetic outline to show the phylogenetic context of the draft genome, involving the 30 closest reference genomes.

In figure 6*e*, we show the phylogenetic context for the 15 (of 25) input contigs whose length achieves the set threshold of 100,000 bp. The contigs are numbered by decreasing length, ranging from B08:1 with length 770,679 bp to B08:15 with length 109,403 bp, respectively. Although the outline indicates that the longest contig B08:1 is very similar to the shown reference genomes, the similarity between contigs and references decreases with decreasing contig length.

Running GTDB-Tk

The frame-shift corrected bins from Arumugam et al. (2019) were classified using the phylogenetic-placement mode of GTDB-Tk (Chaumeil et al. 2019), using the GTDB database R95 version (Parks et al. 2020). We ran the classify_wf workflow with the default settings, using 32 cores both for the main pipeline and for the pplacer program. GTDB-Tk completed the phylogenetic placement of all bins in 26 min. In order to visualize the resulting phylogenetic placements, we opened the Newick-formatted gtdbtk.bac120.classify.tree output file in Dendroscope (Huson and Scornavacca 2012) and manually extracted the relevant subtrees for the bins shown in figure 2 and the Supplementary Material online.

Supplementary Material

Supplementary data are available at *Genome Biology and Evolution* online.

Acknowledgments

D.H.H. acknowledges Catalyst: Leaders funding provided by the New Zealand Ministry of Business, Innovation and Employment and administered by the Royal Society Te Apārangi. The authors acknowledge infrastructural support by the cluster of Excellence EXC2124 Controlling Microbes to Fight Infection (CMFI), project ID 390838134.

Author Contributions

D.B. and D.H.H. conceptualized the project. D.B. and D.H.H. developed the outline algorithm. D.H.H. designed and implemented the software. C.B. and B.C. designed and populated the database. C.B. performed all data analysis and comparisons. D.H.H. and C.B. wrote the original draft of the manuscript and all authors edited the manuscript.

Data Availability

The algorithms are implemented in Java in our program SplitsTree5. Installers for SplitsTree5, and the current database file gtdb-rep-k21-s10000-May2021.db, are freely available here: https://software-ab.informatik.uni-tuebingen.de/download/splitstree5 (last accessed September 16, 2021). The open source is available here: http://github.com/husonlab/splitstree5 (last accessed September 16, 2021). In addition, we provide a Python implementation of neighbor-net and phylogenetic outlines here: https://github.com/husonlab/SplitsPy (last accessed September 16, 2021). No new data were generated or analysed in support of this research.

Literature Cited

Arumugam K, et al. 2019. Annotated bacterial chromosomes from frameshift-corrected long read metagenomic data. Microbiome 7(1):61.

Bloom BH. 1970. Space/time trade-offs in hash coding with allowable errors. Commun ACM. 13(7):422–426.

Bouckaert RR. 2010. DensiTree: making sense of sets of phylogenetic trees. Bioinformatics 26(10):1372–1373.

Bowers RM, et al. 2017. Minimum information about a single amplified genome (MISAG) and a metagenome-assembled genome (MIMAG) of bacteria and archaea. Nat Biotechnol. 35(8):725–731.

Bryant D, Moulton V. 2002. NeighborNet: an agglomerative method for the construction of planar phylogenetic networks. In: Guigó R, Gusfield D, editors. Algorithms in bioinformatics. WABI 2002. Vol. LNCS 2452. Berlin, Heidelberg: Springer. p. 375–391.

Bryant D, Moulton V. 2004. Neighbor-net: an agglomerative method for the construction of phylogenetic networks. Mol Biol Evol. 21(2):255–265.

Buchfink B, Xie C, Huson DH. 2015. Fast and sensitive protein alignment using DIAMOND. Nat Methods. 12(1):59–60.

Chaumeil P-A, Mussig AJ, Hugenholtz P, Parks DH. 2019. GTDB-Tk: a toolkit to classify genomes with the genome taxonomy database. Bioinformatics 36(6):1925–1927.

Dress AWM, Huson DH. 2004. Constructing splits graphs. IEEE/ACM Trans Comput Biol Bioinform. 1(3):109–115.

Franzosa EA, et al. 2018. Species-level functional profiling of metagenomes and metatranscriptomes. Nat Methods. 15(11):962–968.

Holland B, Huber K, Moulton V, Lockhart PJ. 2004. Using consensus networks to visualize contradictory evidence for species phylogeny. Mol Biol Evol. 21(7):1459–1461.

Huson DH, Bryant D. 2006. Application of phylogenetic networks in evolutionary studies. Mol Biol Evol. 23(2):254–267.

Huson DH, et al. 2016. MEGAN Community Edition – interactive exploration and analysis of large-scale microbiome sequencing data. PLoS Comput Biol. 12(6):e1004957.

Huson DH, Rupp R, Scornavacca C. 2010. Phylogenetic networks. Cambridge: Cambridge University Press.

Huson DH, Scornavacca C. 2012. Dendroscope 3 – a program for computing and drawing rooted phylogenetic trees and networks. Syst Biol. 61(6):1061–1067.

Kitts PA, et al. 2016. Assembly: a resource for assembled genomes at ncbi. Nucleic Acids Res. 44(D1):D73–D80.

Matsen FA, Kodner RB, Armbrust EV. 2010. pplacer: linear time maximum-likelihood and bayesian phylogenetic placement of sequences onto a fixed reference tree. BMC Bioinformatics 11(1):538.

Ondov BD, et al. 2016. Mash: fast genome and metagenome distance estimation using MinHash. Genome Biol. 17(1):132.

Ondov BD, et al. 2019. Mash screen: high-throughput sequence containment estimation for genome discovery. Genome Biol. 20(1):232.

Parks DH, et al. 2017. Recovery of nearly 8,000 metagenome-assembled genomes substantially expands the tree of life. Nat Microbiol. 2(11):1533–1542.

Parks DH, et al. 2020. A complete domain-to-species taxonomy for bacteria and archaea. Nat Biotechnol. 38(9):1079–1086.

Pierce NT, Irber L, Reiter T, Brooks P, Brown CT. 2019. Large-scale sequence comparisons with sourmash. F1000Res. 8:1006.

Pruitt KD, Tatusova T, Klimke W, Maglott DR. 2009. NCBI reference sequences: current status, policy and new initiatives. Nucleic Acids Res. 37(Database issue):D32–D36.

Qin Q-L, et al. 2014. A proposed genus boundary for the prokaryotes based on genomic insights. J Bacteriol. 196(12):2210–2215.

Saitou N, Nei M. 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol Biol Evol. 4(4):406–425.

Solomon B, Kingsford C. 2016. Fast search of thousands of short-read sequencing experiments. Nat Biotechnol. 34(3):300–302.

Turakhia Y, et al. 2021. Ultrafast sample placement on existing trees (usher) enables real-time phylogenetics for the sars-cov-2 pandemic. Nat Genet. 53(6):809–816.

Associate editor: Barbara Holland