Estimating microbial (meta)pangenomes with amino acid k-mers OR Protein k-mers enable assembly-free microbial metapangenomics

This manuscript (permalink) was automatically generated from taylorreiter/2021-paper-metapangenomes@8eec3e3 on December 14, 2021.

Authors

- **Taylor Elaine Reiter**
  0000-0002-7388-421X · taylorreiter · ReiterTaylor
  Department of Population Health and Reproduction, University of California, Davis · Funded by Grant XXXXXXXX

- **N. Tessa Pierce-Ward**
  0000-0002-2942-5331 · bluegenes · saltyscientist
  Department of Population Health and Reproduction, University of California, Davis · Funded by NSF 1711984

- **C. Titus Brown**
  0000-0001-6001-2677 · ctb
  Department of Population Health and Reproduction, University of California, Davis

Abstract

Introduction

Short read metagenomic sequencing has expanded our knowledge of microbial communities and diversity (CITE). Many of these insights are attributable to *de novo* assembly and binning, which estimate species-level composite genomes (metagenome-assembled genomes, *MAGs*) from genomes in a sample, capturing unculturable genomes which have expanded the tree of life and and our understanding of microbial metabolism in diverse environments (CITE: Tyson, Hug, Nayfach). Along with these advances, the concept of metapangenomics has arisen as a framework for understanding how sets of genes that occur in closely related MAGs correlate with parameters in the environments in which theyre sampled from (CITE: Delmont, Bing Ma, Hoarfrost). Like pangenome analysis of isolate genomes, metapangenomes reflect the metabolic and ecological plasticity of populations of microbes and give insights into the genes that support specific environmental adaptations (CITE: hoarfrost).

Metapangenomics is reliant on *de novo* metagenome analysis, but both assembly and binning introduce biases into analysis (CITE: CAMI, SGC, Barnum, Maguire, Bickhart). Low coverage or large amounts of variation (SNPs, indels, rearrangements, horizontal gene transfer, sequencing error, etc.) cause assemblers to break contiguous sequences, producing short fragments or unassembled reads that are too short to be binned into MAGs (CITE). These biases disproportionately impact genomic islands and plasmids (CITE: Maguire), hot spots for evolution that support microbial adaptation to changing environments (CITE: Roth?).

To more fully represent the functional potential in metapangenomes, we present an analysis approach that relies on amino acid and reduced alphabet k-mers to accurately estimate microbial pangenomes. K-mers are words of length *k* in DNA or protein sequences. K-mer-based analysis has recently risen in popularity via sketching algorithms that sub sample sequences to facilitate rapid comparisons while maintaining similarity between samples (cite: Rowe). In particular, long nucleotide k-mers preserve similarities between closely related genomes but are brittle to evolutionary distance (CITE: metapalette, sourmash?). By using amino acid k-mers and other reduced alphabets, sequence similarities are preserved across larger evolutionary distances. Combining this approach with accurate open reading frame prediction from short read sequences, this method can be applied without assembly.

Results

We demonstrate ... / summarize results or something here

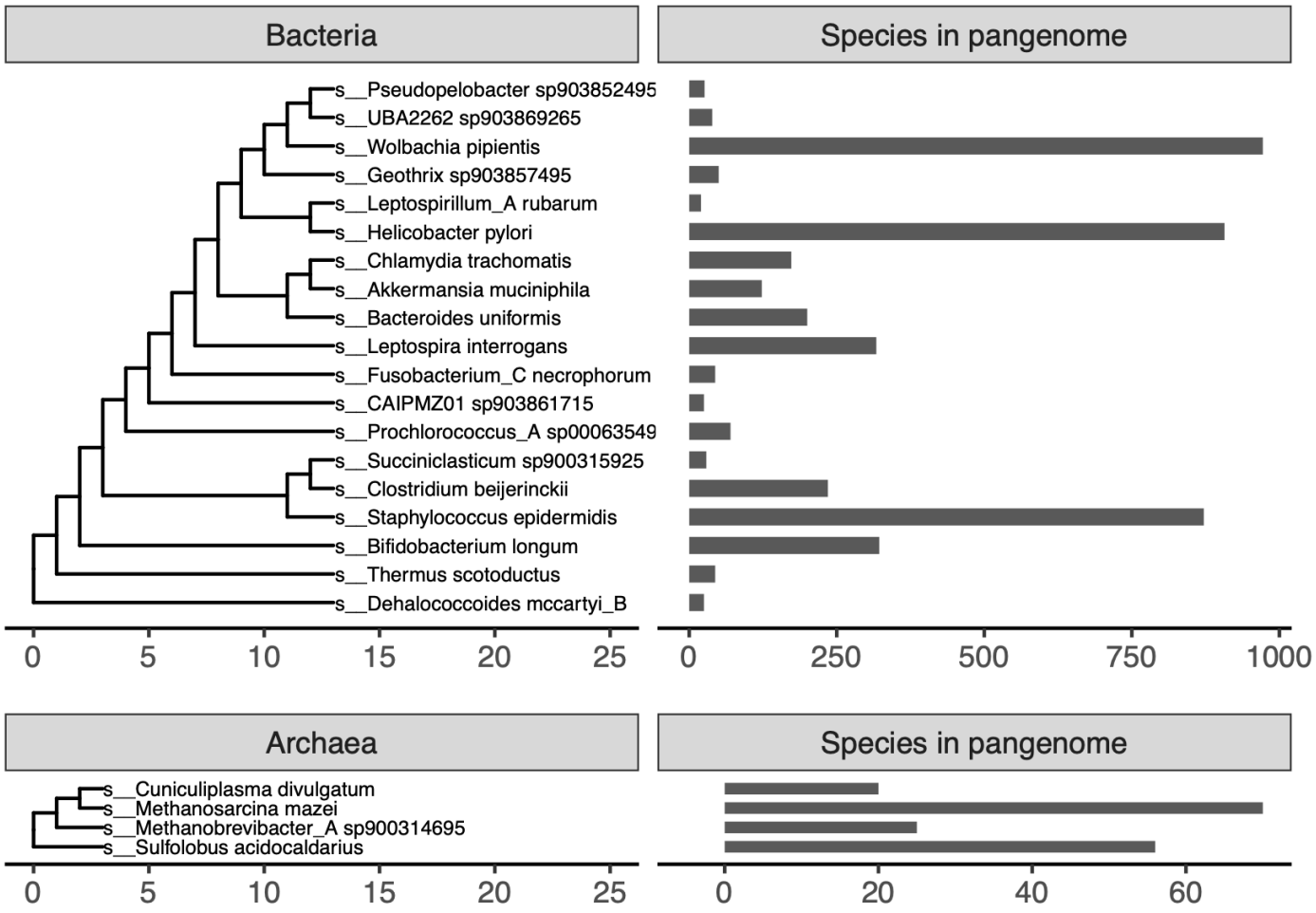

Figure 1: Organisms used in this paper

Reduced alphabet k-mers accurately estimate microbial pangenomes

To determine whether pangenomes could be constructed using amino acid or other reduced alphabet k-mers, we constructed pangenomes from amino acid (k = 7, 8, 9, 10, 11), dayhoff (k = 13, 15, 17), and the hydrophobic-polar (k = 27, 31) encodings and compared these against pangenomes constructed from genes using roary. We constructed pangenomes for 23 species from 23 phyla in the GTDB taxonomy, ranging from 20 to 972 genomes per pangenome (average = 203). For each pangenome, we compared the total number of genes and k-mers and the total number of unique genes and k-mers for each genome. We also tested whether genomes that had similar gene presence-absence profiles had similar k-mer presence-absence profiles using the Mantel test. Performance varied minimally across encodings (**Figure 2**) thus we performed the majority of our analyses using amino acid encodings with a k-mer size of 10.

- [Tessa?]: Do we need to justify selecting a k-mer size of 10 here?
- [Tessa?]: Do I need to show a comparison to a scaled of 1 here? I have it for 12 of the species, and the numbers are similar enough to scaled 100 that I stopped wasting the compute on scaled 1
- [Tessa?]: should I compare this against nucleotide k-mers at all? Bc I think that's the underlying assumption here, nucleotides don't work for this stuff.

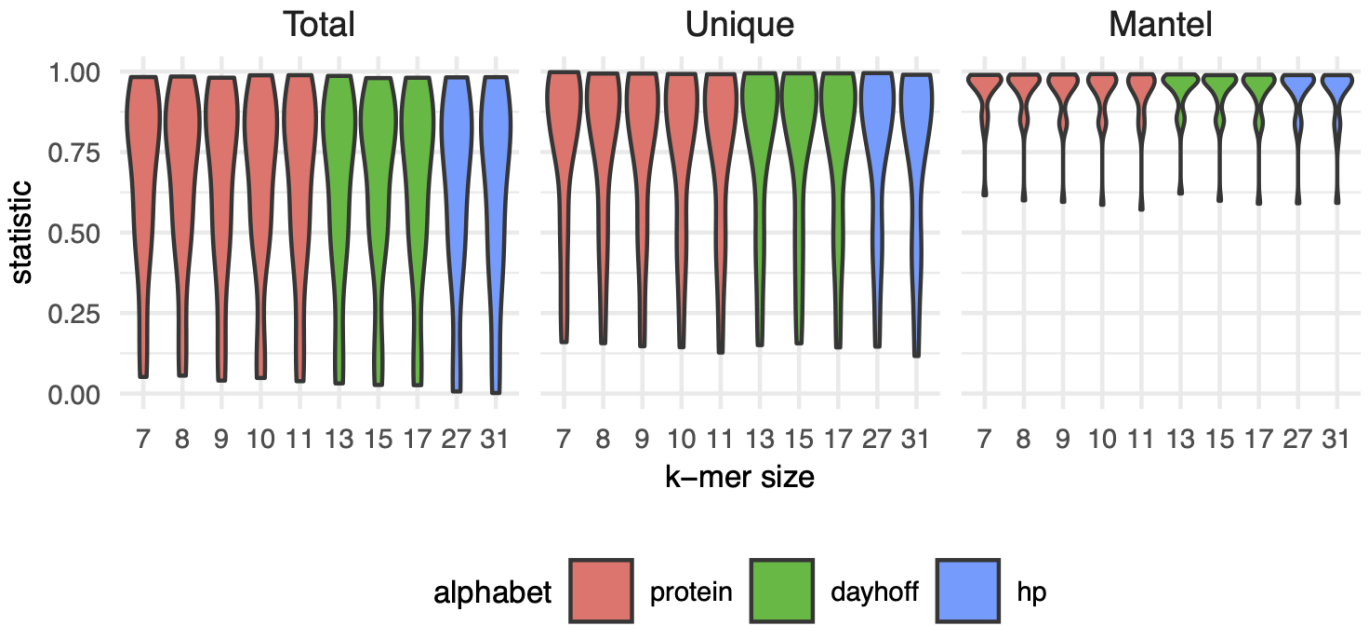

Figure 2: K-size and alphabet do not impact pangenome estimation with k-mers. Violin plots represent the distrubtion of R2 values for linear models (Total, Unique) or statistic values for mantel tests (Mantel). *Total* corresponds to correlation between total number of distinct genes and distinct k-mers in a genome. *Unique* corresponds to correlation between the number of unique genes and unique k-mers in genome. *Mantel* corresponds to a mantel test between the gene and k-mer presence-absence matrices.

Performance across metrics varied dramatically for different pangenomes, with k-mers and genes highly correlated for some pangenomes and not correlated for others. We investigated pangenomes more closely to determine the source of the poor correlations and found that they were caused by the presence of many frameshifted proteins, one of many potential criteria for exclusion of GenBank genomes from RefSeq. For example, *Leptospira interrogans* had an R^2 of 0.12, however 21 genomes contained frameshifted proteins. Removing these genomes increased the R^2 to 0.87 (**Figure 3 A,B**). This trend was consistent across pangenomes, where pangenomes that had at least one genome that was excluded from RefSeq for having many frameshifted proteins had a significantly lower R^2 values between total number of genes and total number of k-mers per genome than pangenomes that did not (Welch Two Sample t-test, estimate = -0.36, p = 0.003) (**Figure 3 C**).

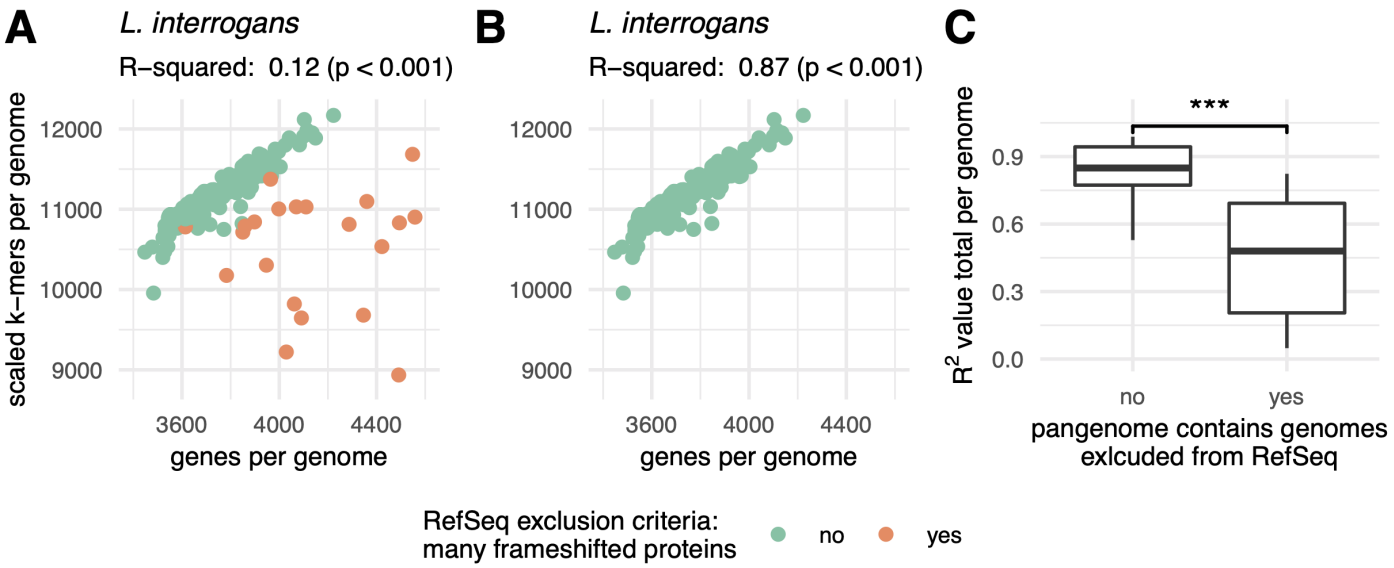

Figure 3: Genomes that are excluded from RefSeq for having many frameshifted proteins reduce similarity between gene- and k-mer-based pangenomes. A, B) Scatterplots of the total number of distinct genes and k-mers per genome for the species *Leptospira interrogans*, where each point represents a single genome in the pangenome.

Removing genomes flagged with RefSeq exclusion criteria “many frameshifted proteins” improves the correlation between these variables. **C)** Boxplot of R^2 values between the total number of distinct genes and k-mers per genome. Pangenomes that contain genomes with the RefSeq exclusion criteria of ‘many frameshifted proteins’ have significantly lower R^2 values

We next investigated whether other pangenome metrics were well correlated between k-mer-based and gene-based methods using pangenomes that did not contain genomes excluded from RefSeq for many frameshifted proteins. For these 13 pangenomes, the percent of k-mers or genes predicted to be part of the core, shell, or cloud genome was strongly correlated (**Figure** 4). We also compared whether pangenomes would be designated as open or closed by calculating the alpha value for the Heaps law model. Alpha was also strongly correlated between gene- and k-mer based pangenomes (**Figure** 4).

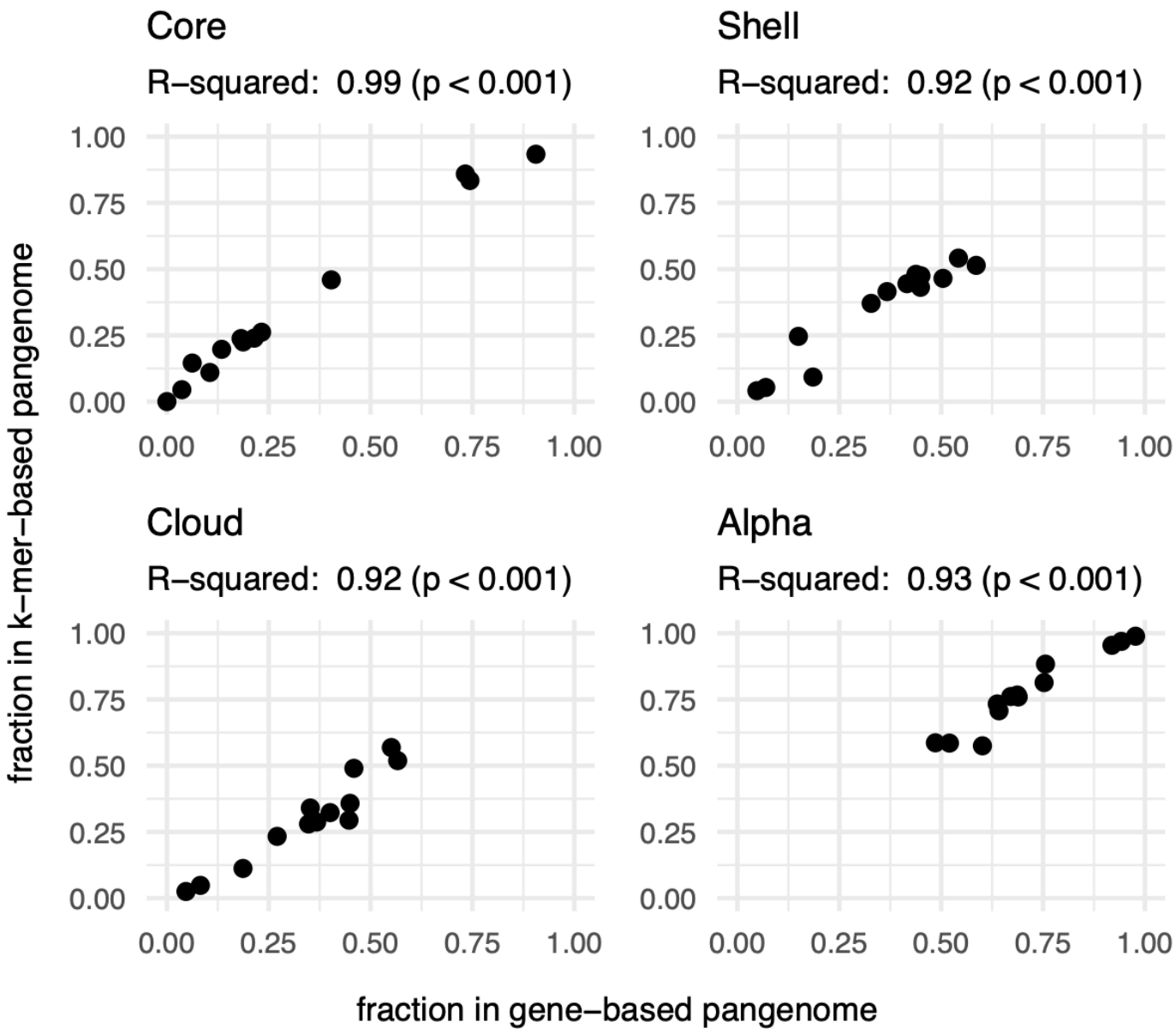

Figure 4: Pangenome metrics strongly correlate between gene- and k-mer-based pangenomes. Pangenome categories core, shell, and cloud refer to genes or k-mers shared between the majority (>95%), some, or singleton genomes in the pangenome. is a value from Heaps law used to estimate whether a pangenome is open or closed.

Lastly, we investigated whether the relationship between the number of k-mers in coding domain sequences in a genome and the number of genes in a genome was constant across diverse phyla represented here. (**Figure** 5)

- [Tessa?]: Is this surprising? I know we discussed this on slack a little, but I guess with the way pangenomes are constructed, maybe its not surprising?

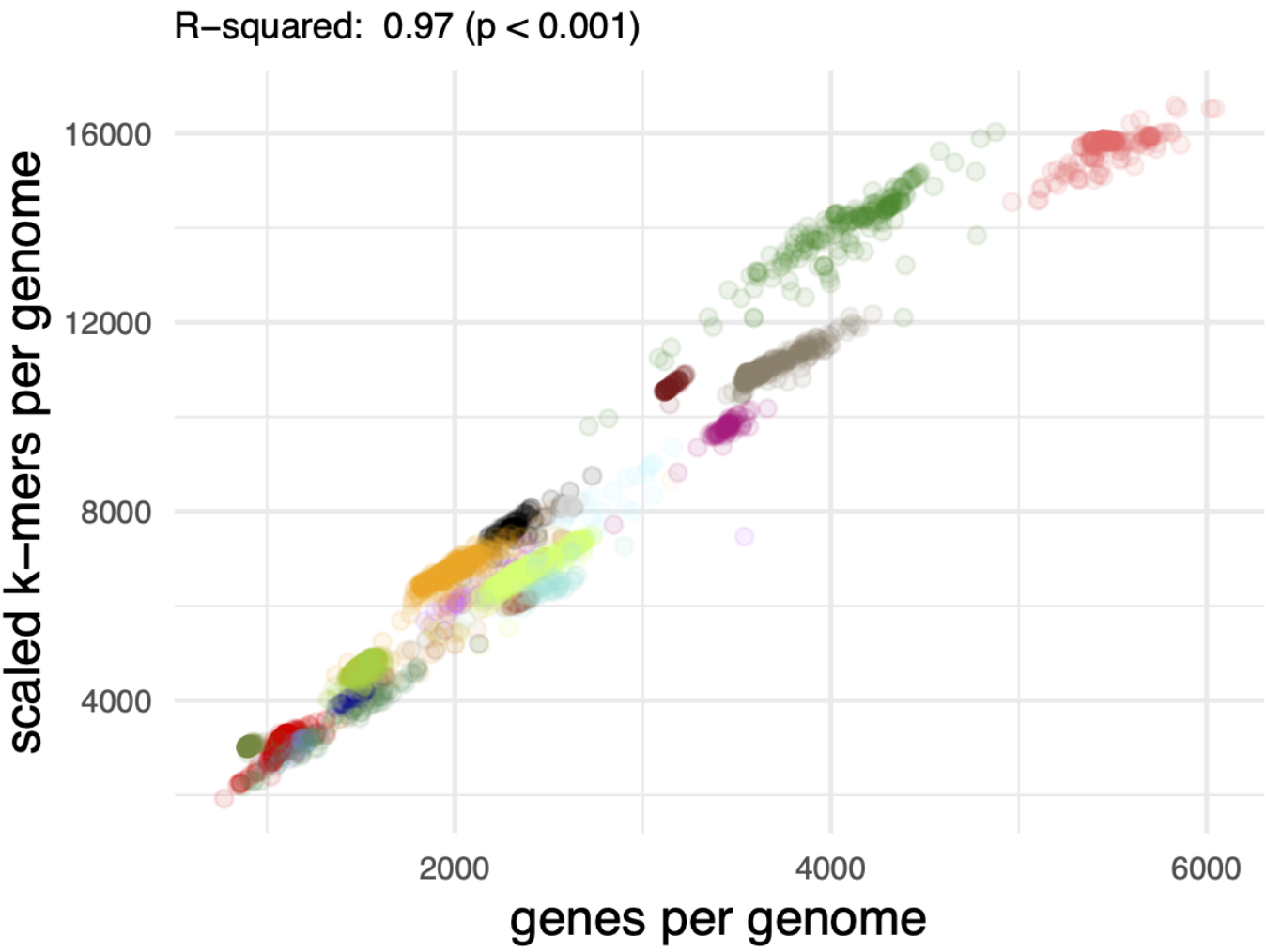

Figure 5: The ratio of total distinct genes per genome to total distinct k-mers per genome is conserved across distantly related species. Each point represents a single genome, and genomes are colored by species.

Jaccard similarity between reduced alphabet k-mers and k-mers in database accurately predicts open reading frames for short sequencing data

Given that protein k-mers can accurately estimate bacterial and archaeal pangenomes, we next sought to determine whether open reading frames could be accurately predicted directly from short sequencing reads, as this would enable pangenome analysis without assembly. We evaluated whether orpheum, a tool recently developed to predict open reading frames in Eukaryotic short reads (CITE), could also perform this this task in bacterial and archaeal sequences. Orpheum predicts open reading frames by comparing reduced alphabet k-mers in six frame translations of short sequencing reads against those in a database (jaccard similarity) and assigns an open reading frame as coding if similarity exceeds a user-defined threshold (CITE). To evaluate orpheum, we constructed a database from all k-mers from coding domain sequences from genomes in GTDB rs202. Using representative genomes from the 23 species above, as well as 20 additional genomes not in the GTDB rs202 database, we simulated short sequencing reads either from coding domain sequences or non-coding sequences in the genome and used these simulated reads to test orpheum.

Orpheum accurately separated coding from non-coding reads for reads that were simulated from genomes in GTDB (**Figure 6 A**).

- Why FN/FP?
  - pseudogenes? fragments? sequencing error?

For reads simulated from genomes not in GTDB, orpheum recovered the majority of coding reads when genomes of the same species were in the database (**Figure 6 A,B**).

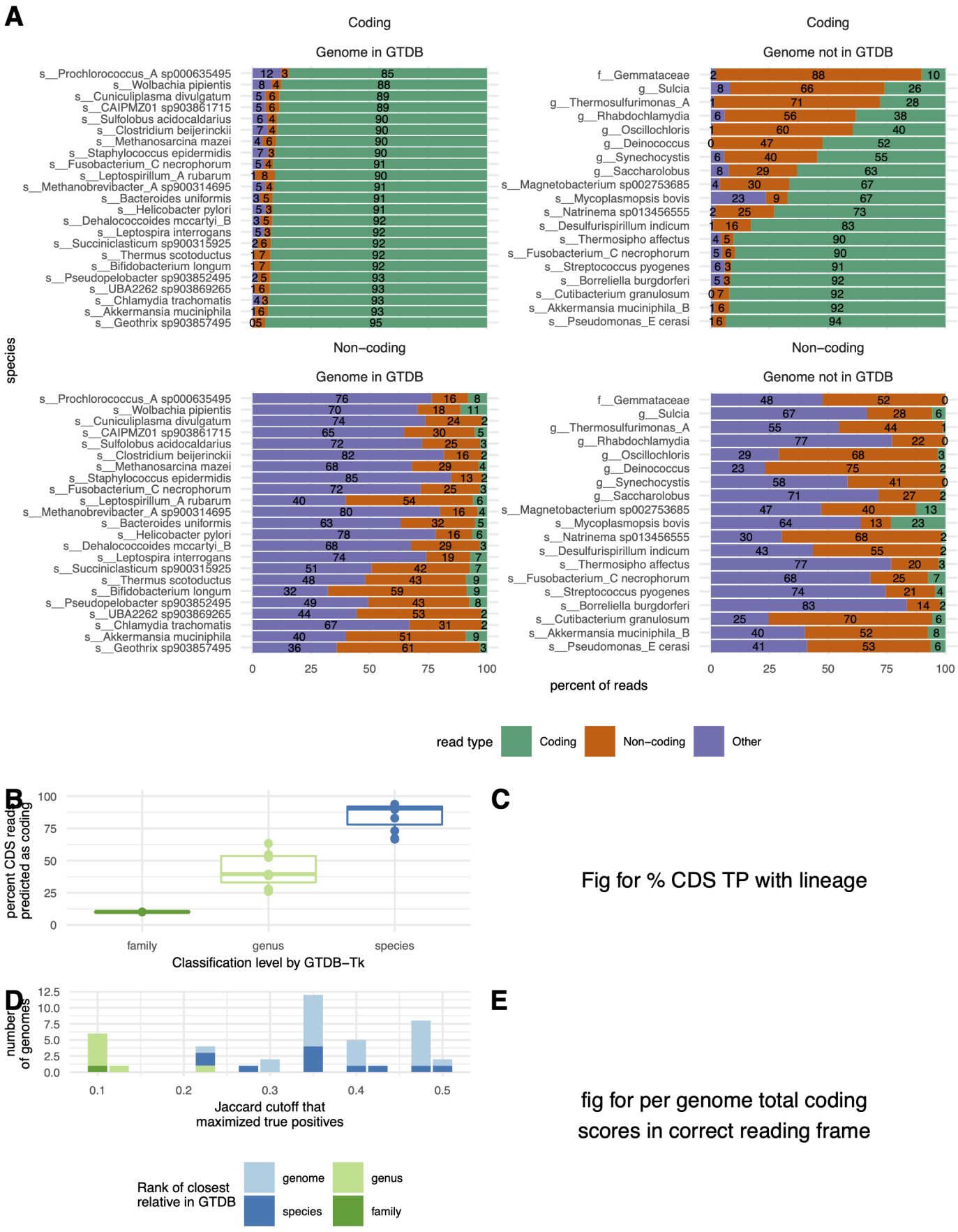

Figure 6: Orpheum correctly assigned short sequencing reads as coding or non-coding and selects the correct open reading frame. A) Percent of simulated coding or non-coding sequences predicted as coding, non-coding, or discarded based on quality metrics (see methods). Genomes are split by those in GTDB and those not in GTDB. **B)** Boxplots of the percent of coding reads that were recovered by Orpheum, separated by the level of taxonomic assignment acheived by GTDB-Tk. Orpheum recovers more coding sequences when there are closely related genomes in the database. **C)** Databases constructed of only closely-related genomes recover the majority of coding sequences, but including increasingly distantly related genomes improves total coding recall. **D)** Barchart of the jaccard similarity

cutoff that maximized the number of true positives for coding and non-coding reads. The best jaccard similarity decreases when there are fewer closely related genomes in the database. **E)** XXX.

OUTLINE

- CDS/nonCDS, in GTDB/not in GTDB results
- what are FP/FN
  - pseudogenes? fragments? “sequencing error”?
- decreasing accuracy with fewer closely related genomes in database
- both genomes of the same species and genomes from distantly related organisms are important for ORF recovery
  - horizontal gene transfer
- lower jaccard similarity to maximize true positives with increasingly distantly related genomes
  - but TP decrease at lower jaccard similarity for genomes that have many genomes of the same species in the DB. Pseudogenes?
- k-mer size, alphabet selection
  - we don’t have any evidence for this yet with these data sets.
- Even with FP/FN rate, comp matrix + mantel test to show we get the same biological results.
- Should/do I have to compare these results against FragGeneScan?

K-mer-based metapangenomics combined with assembly graphs ...

- Do I need to do this? It’s sort of the point, and while both methods are usable on their own for different workflows, it doesn’t all work together without the assembly graph component

- What’s the simplest possible way to do this?

  - Do it for two samples, Hu SB1 and SB2? Could pick a species that has bins in both samples, and that we queried with for the sgc paper.
  - Do it with a time series iHMP person? I have a snakefile that gets half the way there already (https://github.com/taylorreiter/2021-sgc-arg).
  - scoop one of the 54 nbhds I already have for all 605 IBD samples? or some subset of them? I’d really rather not get into that whole schtick, and I don’t have MAGs for those samples (although I guess pasolli or something might...)

- Do I need to compare these results against typical metapangenomics? like do de novo assembly, binning, prokka? etc?

- If I do this – show that k-mers support pangenome visualization by demonstrating. If not, incorporate somewhere else.

Discussion

We present a method to perform reference-free pangenomics. We demonstrate accurate prediction of open reading frames in highly accurate short sequencing reads by comparing amino acid k-mers in all translation frames against a database of k-mers from all known bacterial and archaeal genomes in GTDB (rs202). We then show that pangenome metrics like estimate of core, cloud, and shell pangenome fractions can be accurately estimated with long amino acid k-mers and k-mers from other reduced alphabets. Combining these tools enables pangenome estimation directly from quality controlled short sequencing reads. In the context of metagenomes, these approaches enable

metapangenome estimation without the need to *de novo* assemble and bin sequences, eliminating common sources of lost sequencing variation (cite spacegraphcats).

The combination of these approaches is potentially most useful in the context of analyzing metagenome assembly graphs. Assembly graphs like compact de Bruijn graphs (cDBG) capture all sequences in a metagenome, including sequences with high strain variation or low coverage, which may not be captured by other analysis methods. A targeted query of an assembly graph, for example with a metagenome-assembled genome bin, can recover all sequencing reads in a metagenome that originate from all genomes of the same species (cite spacegraphcats). While recovering these reads and assigning their taxonomic identity through graph queries is useful, many of the recovered reads cannot be assembled due to prolific sequencing variation attributable to strain diversity in the original microbial community. Yet, the sequences represented by these un-assembleable reads often encode functional potential, some of which may be key to a microorganisms functioning within its ecosystem (cite sumner paper?; metachercant). The approaches presented in this paper enable these sequences to be represented in pangenome estimation.

Long read sequencing of microbial communities stands to improve many of these challenges, particularly as lineage-resolved methods become mainstream (cite bickhart et al.). Even as long read technologies improve, short read sequences continue to better capture strain diversity from a community (Cite Maureen?). Even with long read references from the same community, many of these short reads do not map and do not assemble (cite Maureen). The approaches presented here will allow these sequences to be included in pangenome estimation.

Practically, open reading frame prediction with orpheum can be executed on microbial illumina short read data sets. The RAM used to run orpheum is dictated by the database size, as the database is loaded into to memory while its running. The GTDB rs202 nodegraph was 94 GB, and the RAM required to run orpheum never exceed 97GB, which makes database distribution and orpheum execution available on high performance compute clusters and other remote computers. To reduce ram, this data structure could be improved XXX.

We demonstrated that orpheum is better able to predict open reading frames in genomes that have species-level representatives in the GTDB database. To asses whether this criteria is satisfied by a query genome without performing genome assembly, we recommend sourmash gather. Sourmash gather will estimate the fraction of sequencing reads in a genome or metagenome that match to a genome in GTDB by comparing long nucleotide k-mers in the query against those in the database (cite gather paper). Alternatively, the tool SingleM could be used to perform this task. SingleM estimates the taxonomic composition of sequencing reads by identifying fragments of single copy marker genes in short reads and comparing them against a database of taxonomically labelled sequences.

While it is necessary for the database to have a species-level representative for orpheum to achieve high ORF recall for a given query, it is not sufficient. This likely reflects similarity in core genomes for closely related organisms, but prolific horizontal gene transfer between both closely and distantly related organisms. Including genomes from increasingly distant taxonomic ranks in the database added XX additional true positives.

Comparison between euks? Need to read orpheum paper.

PANMER discussion

- sourmash signature generation is rapid.
- Exact matching scales (linearly?). May enable running on very large collections of genomes.
- Exact matching of k-mers enables additions of new species without having to rerun everything.

- Exact matching also allows direct comparisons to distantly related organisms. Unified framework for genome comparisons even when organisms are distantly related.
- scaled is handy parameter to potentially enable even larger comparisons
- sacrifice function – annotating k-mers with function is good future work.

Other points

- While the number of genes per genome is increased for genomes with this exclusion criteria, there is no commiserate increase in the number of k-mers observed. This suggests that the number of k-mers in a genome could be used to predict the expected range of predicted genes in a genome, and could be potentially used a quality control metric for annotated genomes.
- While developed for the metapangenomics space, this study demonstrates that k-mer-based pangenomes will also work in isolate genomes. Given that building k-mer sketches and exact matching of k-mers between genomes is fast, this provides an alternative approach for building pangenomes.
- De novo metagenome analysis probably dramatically improves ORF prediction because of the inclusion of these genomes in GTDB.

Methods

All code is available at github.com/taylorreiter/2021-panmers and github.com/taylorreiter/2021-orpheum-sim.

Selection of benchmarking species for pangenome analysis

We selected a species representative for each of the 23 phyla in GTDB rs202. To select representative species, we first filtered species with fewer than 20 representatives and greater than 1000 representatives. While this approach scales beyond 1000 genomes, we elected to benchmark smaller sets to iterate over the potential parameter space more quickly. Of species remaining after filtering, we selected the species within each phyla that had the largest number of genomes. We downloaded these genomes from GenBank. Species names are in table XXX.

Calculating the gene-based pangenome with roary

To calculate the gene-based pangenome, we first annotated each genome using prokka with the `--metagenome` flag. We then used the resulting GFF annotations files to calculate the pangenome with roary using default settings.

Calculating the k-mer based pangenome with sourmash

To calculate k-mer based pangenomes, we used sourmash `sketch` to generate signatures from the prokka-predicted amino acid sequences (`.faa` files). We used the protein alphabet (k = 7, 8, 9, 10, 11), dayhoff alphabet (k = 13, 15, 17), and the hydrophobic-polar alphabet (k = 27, 31). All signatures were calculated with a scaled value of 100. The scaled parameter controls the fraction of the total k-mers represented by the sketch; a scaled value of 100 indicates that 1/100th of the distinct k-mers in a genome were included in each sketch. We converted signatures from json format into a genome x hash presence-absence matrix.

Correlating gene-based and k-mer based pangenomes

Using the presence-absence matrices for the gene-based and k-mer-based pangenomes, we correlated total genes/k-mers observed per genome and total unique genes/k-mers observed per genome for each species. We used the `rowSums()` function in R to determine the number of genes/unique genes per matrix, then used the `lm()` function with default parameters to correlate the values. We also used the Mantel test to determine whether genomes that were most similar in the gene presence-absence matrix were also most similar in the k-mer presence-absence matrix. We used the `mantel()` function in the R vegan package to perform this test. We used distance matrices calculated with the `dist()` function using the parameter `method = "binary"` as input to the mantel test.

Generating standard pangenome metrics with pagoo

The pagoo R package provides functions to analyze bacterial pangenomes. We used this package to generate standard pangenome metrics and visualizations. These metrics are based on the presence-absence matrices generated above and include calculation of the core, shell, and cloud genome sizes and estimation of the alpha value in Heaps law for estimation of pangenome openness.

Augmenting benchmarking species set to include genomes not in GTDB for open reading frame prediction

We next generated a benchmarking data set for open reading frame prediction. We selected a genome from each of the 23 species evaluated above, choosing the GTDB rs202 representative genome for each species. Given that open reading frame prediction relies on a database, and we used k-mers in GTDB rs202 to generate this database, we also wanted to select genomes that were not in GTDB to evaluate this method. We determined the bacterial and archaeal genomes that were added to RefSeq after the construction of GTDB rs202 (April 2021-November 2021). From this set, we selected a representative genome from each of the distinct NCBI phyla represented among these genomes, 20 in total. Genome accessions are recorded in Table XXX. We then ran GTDB-tk on these genomes to predict the GTDB taxonomy of each.

Simulating coding domain sequence and non coding domain sequence reads with polyester

We next created a labelled data set of simulated reads that were generated from either coding domain sequences (CDS) or non-coding regions within each genome. We annotated the genomes with bakta to produce CDS ranges, and used polyester to simulate reads from CDS or non-coding regions. We used the default short read error profile within polyester.

Determing short read open reading frames with orpheum

We used the orpheum tool to predict open reading frames from simulated short reads. Orpheum was developed to predict open reading frames in short RNA-seq reads from Eukaryotic organisms without a reference genome or transcriptome sequence. Orpheum perform six-frame translation on nucleotide sequencing reads, calculates k-mers in an amino acid, dayhoff, or hydrophobic-polar encoding at the designated k-mer length, and then estimates the jaccard similarity between k-mers in each translation frame and a database. It then selects all open reading frames based on a jaccard similarity threshold, and returns those reads as translated amino acid sequences. Open reading frames are excluded if they contain stop codons, low complexity sequences, or if the read is too short to perform translation. Reads are designated as non-coding if they don't reach the jaccard similarity threshold and are not excluded for other reasons.

We constructed a database from GTDB rs202 using sourmash XXX and using a k-mer size of 10. + [**Tessa?**] any relevant details would be very helpful :)

References