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@5a75202 on December 21, 2021.

Authors

- **Taylor E. 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

- **Luiz Irber**
  0000-0003-4371-9659 · luizirber · luizirber
  Graduate Group in Computer Science, UC Davis; Department of Population Health and Reproduction, University of California, Davis

- **Olga Borisovna Botvinnik**
  0000-0003-4412-7970 · olgabot · olgabot
  Data Sciences Platform, Chan Zuckerberg Biohub

- **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 [1,2,3]. INTRODUCE REFERENCE BASED METAGENOMICS, DE NOVO METAGENOMICS.

Along with these advances, the concept of metapangenomics has arisen as a framework for understanding how sets of metagenome-derived genes that are attributable to a group of organisms correlate with parameters in the environments in which they are sampled from [4,5,6]. Metapangenomic methods borrow heavily from pangenome analysis. Pangenomes comprise all genes found within a group of organisms and reflect teh metabolic and ecological plasticity of that group (CITE). The pangenome is divided into core and accessory genes, where core genes are shared by almost all members in teh group and accessory genes are not. Core genes often encode primary metabolic function or other functions necessary for a group to live in a given environment (CITE), while accessory genes encode functions that facilitate adaptation to changing environments (CITE: Roth). The size of the pangenome reflects the diversity of the organisms in a pangenome (population size, number of organisms sampled) as well as the ability of those organisms to adapt to different niches (CITE: Tettelin 2005). Open pangenomes are those which increase indefinitely in size when adding new genomes, while closed pangenomes do not.

While pangenomes are traditionally inferred from isolate genomes, metapangenomics extends the ecological framework of pangenomics to metagenomes. Metapangenomes reflect the metabolic and ecological plasticity of populations of microbes and give insights into the genes that support specific environmental adaptations [5]. This is achieved either through the application of pangenome methods to metagenome assembled genomes (MAGs) (CITE), or by mapping metagenomes against isolate-inferred pangenomes (CITE: Delmont). Both methods give valuable insight into the presence nad distribution of functional content in natural microbial communities, but either may introduce biases associated with unknown sequencing content (CITE: Segata, unknown). MAGs are often incomplete or unrecoverable due to low sequencing coverage or large amounts of variation (SNPs, indels, rearrangements, horizontal gene transfer, sequencing error, etc.), both of which cause short read assemblers to produce unbinnable short contiguous sequences. Unbinned sequences are disproportionately comprised of genomic islands and plasmids [7], hot spots for evolution that support microbial adaptation to changing environments [8]. In contrast, read mapping against isolate-inferred pangenomes may miss functional content present in the metagenome but missing from references, especially for species under represented or unrecorded in reference databases.

These issues are not exclusive to metapangenome inference, and many recently developed analysis strategies overcome some of these baises. These techniques largely rely on k-mers, words of length *k* in DNA or protein sequences. Metagenome k-mer profiles contain all sequences in a metagenome, including those which may not assemble or bin, or which aren't in reference databases. Long k-mers are also taxonomy-specific, where increasing k-mer length leads to sub-species discriminatory power [9] (CITE: tessa). These properties have popularized the use of k-mers for metagenome analysis, primarily through lightweight sketching and compact de Bruijn assembly graphs (cDBGs). Lightweight sketching facilitates fast and accurate sequence comparisons between potentially large data sets through random but consistent sub-sampling (CITE: Mash, gather paper). cDBGs maintain connectivity between k-mers and organize them into species-specific neighborhoods (CITE: barnum, sgc).

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 estimate microbial (meta)pangenomes. In order to derive these k-mers directly from shotgun metagenome reads, we demonstrate the accuracy of a tool called orpheum for open reading frame prediction from short sequencing reads. We demonstrate the application of the method to species present over time in a time series metagenomes from a human gut microbiome. We use assembly graph genome queries to retrieve species-specific reads from the metagenome, predict open reading frames from those reads using orpheum, and build a metapangenome using protein k-mers. This approach for metapangenome estimation is minimally reliant on reference databases and is assembly-free.

Results

We demonstrate … / summarize results or something here

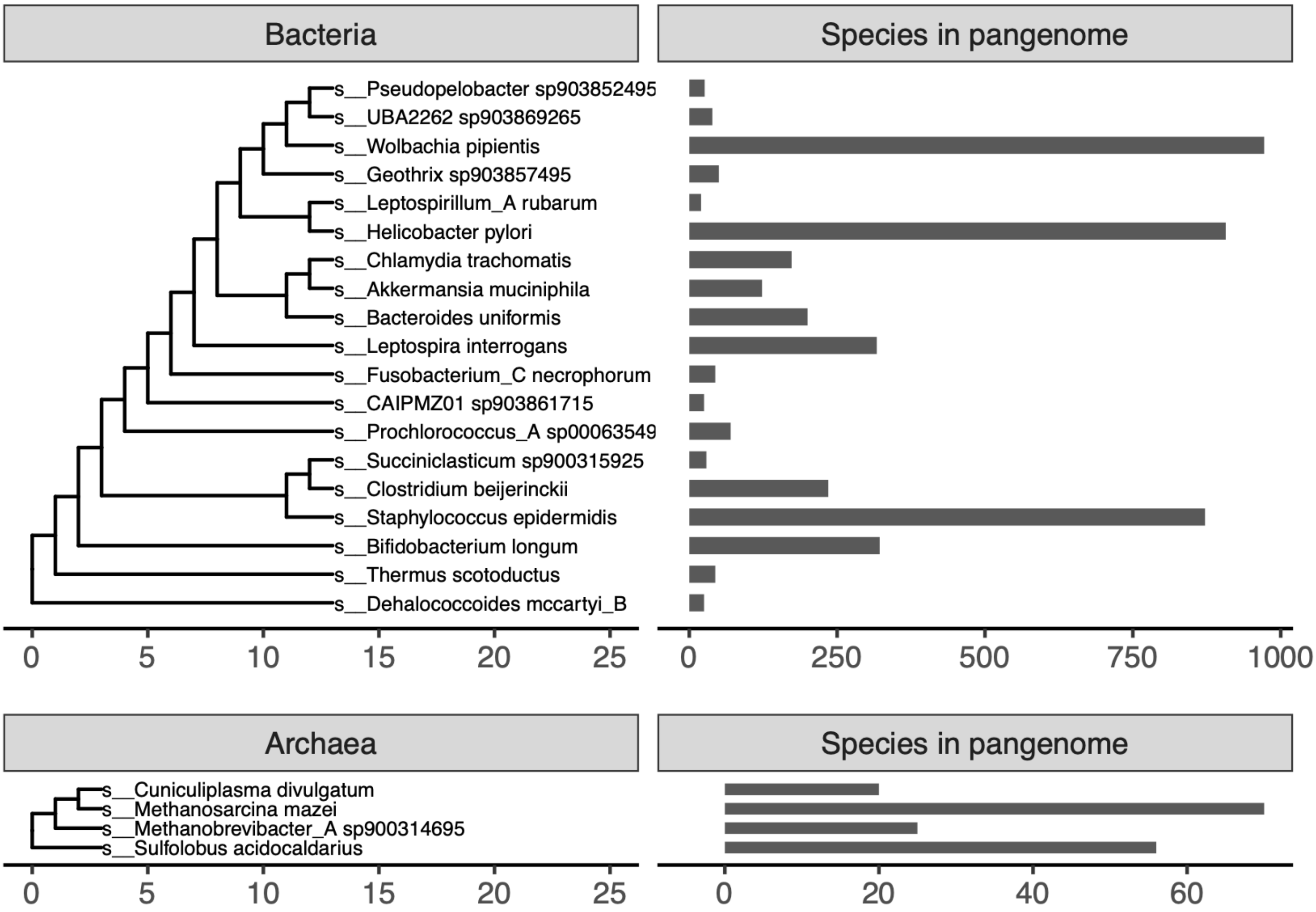

Figure 1: Organisms used in this paper

Reduced alphabet k-mers accurately estimate microbial pangenomes

Pangenomes from isolates are typically built by assembling each isolate genome and predicting genes (open reading frames), clustering gene sequences from all genomes into a non-redundant set, and estimating the presence/absence or abundance of each gene in each genome. To determine whether bacterial and archaeal pangenomes could be constructed from reduced alphabet k-mers, we compared pangenomes estimated from genes against those estimated from k-mers (amino acid, dayhoff, and hydrophobic-polar). We compared pangenomes from 23 species belonging to 23 phyla in the GTDB taxonomy (CITE), with pangenome size ranging from 20-972 genomes (mean = 203 genomes). For each pangenome, we compared the total number of genes and k-mers and the total number of unique genes and k-mers within each genome. We also tested the similarity of presence/absence profiles between pangenomes constructed with different methods using the Mantel test.

For these three metrics, performance varied minimally across encodings and k-mer sizes (**Figure 2**). This is likely because the genomes of the same species are closely related, so any reduced alphabet k-mer is sufficient to overcome minor variations such as those introduced by third base pair wobble (CITE). Given that neither encoding nor k-mer size impacted these performance metrics, we selected protein k-mers with k = 10 to complete the rest of our analysis. Protein k-mers of length 10 have recently been shown to perform well for comparisons across variable taxonomic distances (CITE: TESSA).

- Do I need to discuss scaled at all? Or 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
- 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. Tessa stuff sort of already shows this.

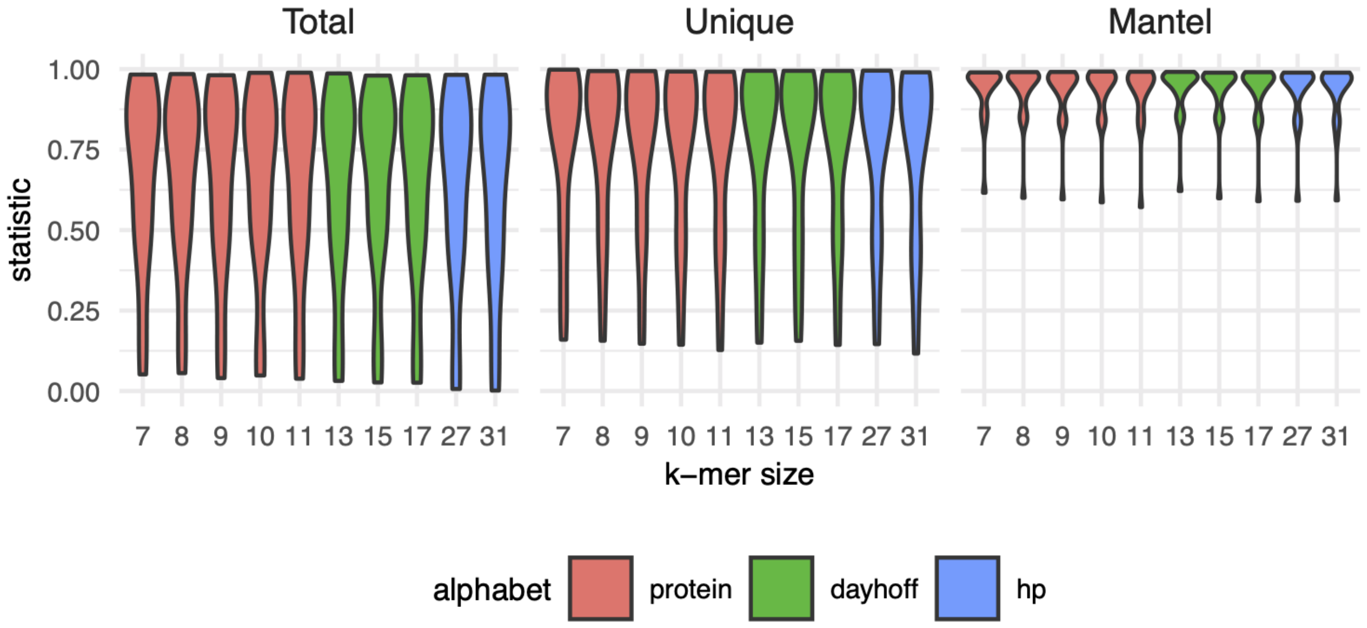

Figure 2: K-size and alphabet do not impact pangenome estimation with k-mers. Violin plots representing the distribution of R^2 values for linear models (Total, Unique) or statistic values for mantel tests (Mantel) calculated for each pangenome. *Total* corresponds to correlations between the total number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

While performance across metrics was similar across encodings and k-mer sizes, it varied dramatically for different pangenomes: k-mers and genes highly correlated for some pangenomes and not correlated for others (**Figure 2**). 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 between the total number of genes and k-mers in genomes in the pangenome, however 21 of 317 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 k-mers per genome than pangenomes that did not (Welch Two Sample t-test, estimate = -0.36, p = 0.003)

(**Figure 3 C**). Other RefSeq exclusion criteria did not impact the correlation between the total genes and k-mers per genome for a given pangenome.

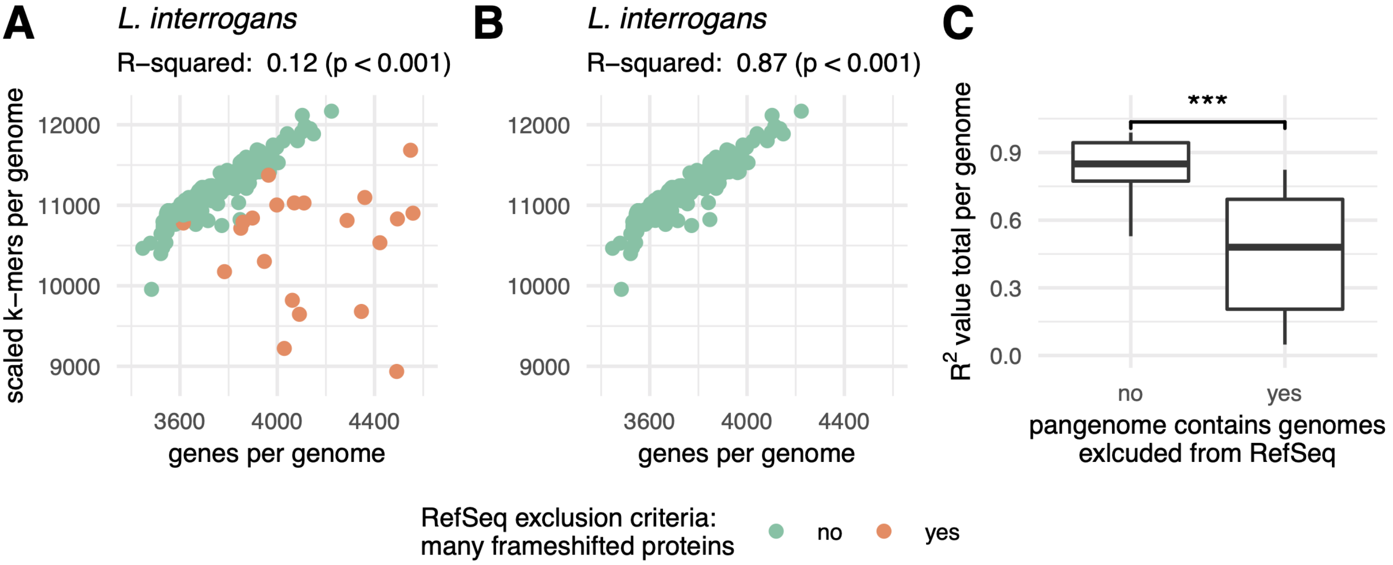

Figure 3: Genomes that are excluded from RefSeq for having many frameshifted proteins reduce similarity between gene- and k-mer-based pangenomes. A, B) Scatter plots of the total number of distinct genes and k-mers per genome for the species *Leptospira interrogans*, where each point represents a single genome in the pangenome. Removing genomes flagged with RefSeq exclusion criteria "many frameshifted proteins" improves the correlation between these variables. **C)** Box plot 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 having 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 values were strongly correlated between gene- and k-mer based pangenomes (**Figure 4**).

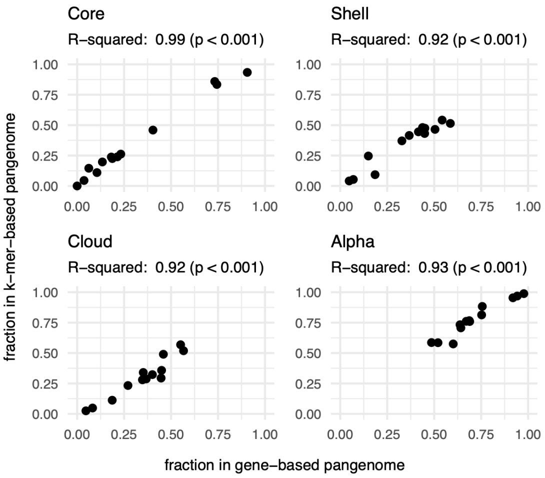

Figure 4: Pangenome metrics strongly correlate between gene- and k-mer-based pangenomes. Pangenome categories core, shell, and cloud refer to genes or k-mers shared between the majority (>95%), some, or singleton genomes in the pangenome. Alpha is a value from Heaps law used to estimate whether a pangenome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

Taken together, these results show that reduced alphabet k-mers can accurately estimate pangenomes from bacterial and archaeal genomes.

Jaccard containment between reduced alphabet k-mers and k-mers in databases accurately predicts open reading frames in short sequencing reads

We next sought to determine whether open reading frames could be accurately predicted directly from short sequencing reads, as this would enable k-mer-based pangenome analysis from short reads. We evaluated whether orpheum, a tool recently developed to predict open reading frames in Eukaryotic short reads [10], could also perform 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 containment) and assigns an open reading frame as coding if containment exceeds a user-defined threshold [10]. To evaluate orpheum, we constructed a database containing all k-mers in coding domain sequences from genomes in GTDB rs202. Using representative genomes from the 23 species above, as well as 20 additional RefSeq genomes not in the GTDB rs202 database, we simulated short sequencing reads either from coding domain sequences or non-coding sequences and used these reads to test orpheum.

Using default parameters, orpheum accurately separated coding from non-coding reads when reads were simulated from genomes in GTDB (**Figure 5 A**). 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 5 A,B**). Accuracy decreased with increasing taxonomic distance between the query genome and the closest relative in the database (**Figure 5 B**).

For genomes that had at least species-level representatives in GTDB, the largest source of error was non-coding reads being predicted as coding (**Figure 5 A**). We hypothesized that these reads originated from pseudogenes as these sequences would likely not be annotated as coding in the genomes from which the reads were simulated from, but may retain some k-mers contained in the database. To assess this hypothesis, we used annotation files produced by the NCBI Prokaryotic Genome Annotation Pipeline (PGAP), which annotates pseudogenes, for the 23 genomes for which these files were available [11,12]. On average, 12.4% (SD = 13.8%) of non-coding reads that were predicted to be coding fell within pseudogenes annotated by the PGAP pipeline. We then BLASTed a subset of the remaining non-coding read that were predicted to be coding against the NCBI nr database. All reads we investigated had at least one mach at 100% identity to protein sequences in the database, suggesting our test genomes contained additional pseudogenes not annotated by PGAP, or that the software we used to predict open reading frames missed some coding sequences (see Methods). Because this method of open reading frame prediction cannot distinguish pseudogenes, it may not be appropriate for species with many pseudogenes.

Some coding sequences were also predicted to be non-coding. We hypothesized that this was caused by sequencing error introduced into the simulated reads. We mapped the simulated reads against the coding domain sequences from which they were derived and calculated mapping error rates. While all reads mapped, the error rate was higher for reads that were predicted to be non-coding than those predicted to be coding (Welch Two Sample t-test, estimate = 0.00523, $p < 0.001$).

Protein k-mers from predicted open reading frames in the simulated short sequencing reads recapitulated similarity between genomic coding domain sequences. We estimated the jaccard similarity between genomes using protein k-mers ($k = 10$) from annotated coding domain sequences, and compared this against jaccard similarity between genomes using protein k-mers from predicted open read frames in the simulated short sequencing reads. Genomes that were most similar in one matrix were also most similar in another matrix (Mantel statistic = 0.9975, $p < 0.001$). The average similarity among all pairwise comparisons for the coding domain sequences was 2.6%, and this decreased to 2.5% when using the open reading frames predicted from reads. This demonstrates that information recovered from open reading frame prediction from short read is similar to that derived directly from the genome sequence.

The majority of predictive capability originated from species-level databases. We performed ORF prediction using just species-level databases for genomes that had at least a species-level representative in GTDB, and compared this against ORF prediction using the full GTDB database. On average, there was no change between the percent of reads derived from coding domain sequences when a species-level database was used versus when all of GTDB was used to predict open reading frames (**Figure 8**).

Decreasing the jaccard containment threshold increased the sensitivity and specificity of ORF prediction when there are no closely related genomes in the database (**Figure 5 C, Table 1**). The jaccard containment threshold controls the final prediction of coding vs. non-coding, as well as the the number of open reading frames which a read is translated into. On average, increasing the rank of the closest taxonomic relative in the database by one taxonomic level decreased the optimal jaccard containment threshold by 0.13.

Table 1: Jaccard containment thresholds that maximize the Youden's index depending on the taxonomic rank of the closest relative in GTDB.

jaccard threshold	closest rank	mean sensitivity	mean specificity	mean Youden's index
0.47	genome	0.988	0.971	0.959
0.39	species	0.941	0.961	0.902
0.17	genus	0.790	0.862	0.653
0.07	family	0.593	0.878	0.471

Overall, these results show that open reading frames can be accurately determined from short sequencing reads when closely related organisms have been previously sequenced.

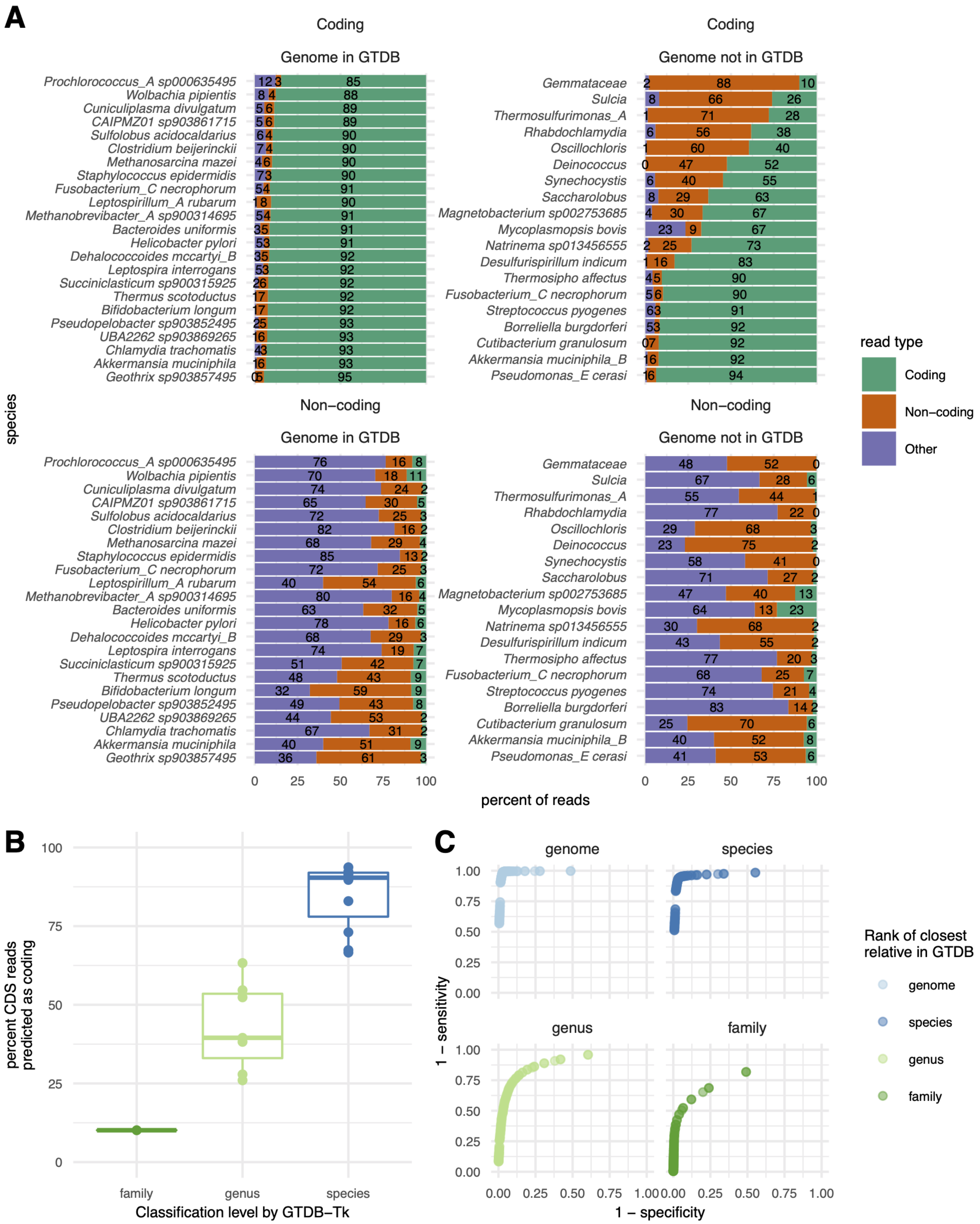

Figure 5: Orpheum correctly assigned short sequencing reads as coding or non-coding and selects the correct open reading frame. A) Percent of simulated coding or non-coding sequences predicted as coding, non-coding, or discarded based on quality metrics (see methods). Genomes are split by those in GTDB and those not in GTDB. Genomes not in GTDB are labelled by taxonomic assignment from GTDB-tk. Predictions were made using default parameters (jaccard containment = 0.5). **B)** Boxplots of the percent of coding reads that were recovered by Orpheum, separated by the level of taxonomic assignment achieved by GTDB-Tk. Orpheum recovers more coding sequences when there are closely related genomes in the database. **C)** Receiver operating curves for the jaccard containment thresholds. Curves are separated by the level of taxonomic assignment achieved by GTDB-Tk, and values are averaged across all genomes that fell within those categories. The best jaccard threshold decreases when there are fewer closely related

genomes in the database. **D)** Databases constructed of only closely-related genomes recover the majority of coding sequences, but including increasingly distantly related genomes improves total coding recall.

- Should/do I have to compare these results against FragGeneScan?

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

Given that amino acid k-mers accurately estimated pangenomes, and that the correct open reading frame could be predicted from short sequencing reads, we next combined these approaches to perform metapangenome analysis from short read shotgun metagenomes. We used 12 metagenomes from a single individual that were sampled over the course of a year by the Integrated Human Microbiome Project (iHMP) [13]. The individual was diagnosed with Crohn's disease, a sub type of inflammatory bowel disease characterized by inflammation along the gastrointestinal tract. The individual received three courses of antibiotics over the year and each course was separated by weeks without antibiotics (**Figure 6**).

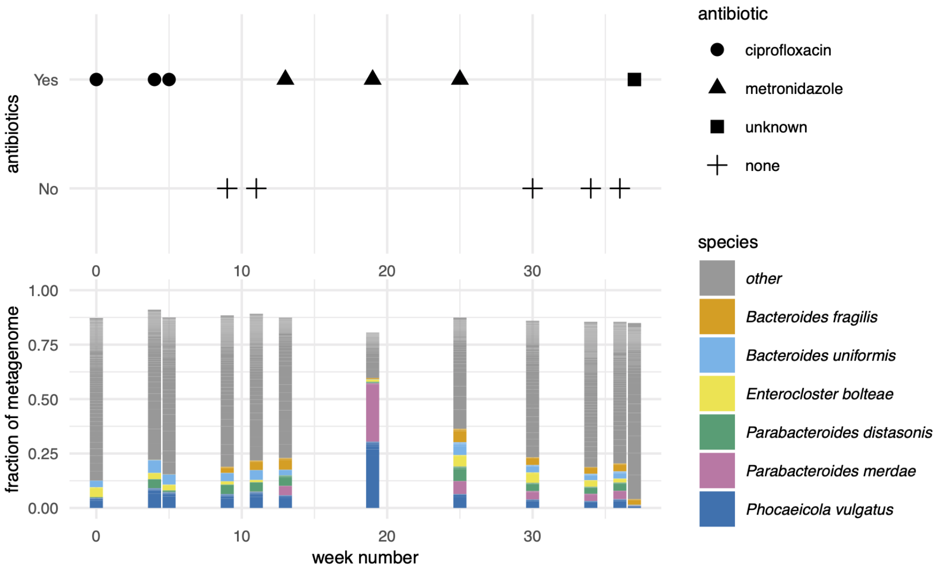

Figure 6: Antibiotic courses and corresponding gut microbiome profiles for a single individual with Crohn's disease. Fractional abundances are colored by species, with only the six species that accounted for greater than 2% of all metagenome reads displayed.

We estimated the metapangenome for each species that was detected in all 12 metagenomes and that accounted for at least 2% of reads across metagenomes, for a total of six metapangenomes (**Figure 6**). To obtain all sequencing reads that originated from genomes of these species, we performed assembly graph genome queries [14]. Assembly graphs contain all sequences in a metagenome, and assembly graph queries return sequences in the metagenome that are either in the query or nearby to the query in the graph. Assembly graph genome queries return sequencing reads that originate from genomes in the metagenome that have as little as 0.1 jaccard similarity (approximately 93% average nucleotide identity (ANI) (CITE: TESSA)) to the query genome [14]. After retrieving reads in this way, we predicted open reading frames using orpheum. We used species-level databases as these were successful in the context of isolate genomes not in the database (see above) and because they would be more likely to filter out reads beyond the species boundary (95% ANI [15]) that were returned by assembly graph queries. Using the predicted amino acid sequences, we built metapangenomes for each of the six species (**Figure 7**).

Unlike isolate genomes, metagenomes may contain a fraction of an organisms genome if the metagenome was not sequenced deeply or if an organism was rare. To calculate the core, shell, and cloud fractions and to estimate the openness of the metapangenome, we removed samples with fewer than 10,000 k-mers (**Table 2**).

Table 2: Metapangenome estimates for each species. *n* designates the number of metagenomes used to estimate the total, core, shell, cloud, and alpha values.

species	n	total	core	shell	cloud	alpha
Bacteroides fragilis	7	24819	56.3%	11.3%	32.4%	0.76
Bacteroides uniformis	9	32197	38.0%	22.3%	39.7%	0.73
Enterocloster bolteae	4	23620	55.8%	18.3%	25.9%	0.66
Parabacteroides distasonis	7	25789	42.4%	30.9%	26.8%	0.74
Parabacteroides merdae	6	19985	63.2%	9.6%	27.1%	0.82
Phocaeicola vulgatus	11	41005	30.3%	20.4%	49.2%	0.65

```
          species  n total  core shell cloud     alpha core_pct shell_pct
cloud_pct
```

1 Bacteroides fragilis 7 24819 13983 2797 8039 0.7560484 56.3 11.3 32.4 2 Bacteroides uniformis 9 32197 12238 7188 12771 0.7267620 38.0 22.3 39.7 3 Enterocloster bolteae 4 23620 13189 4324 6107 0.6625238 55.8 18.3 25.9 4 Parabacteroides distasonis 7 25789 10922 7960 6907 0.7352314 42.4 30.9 26.8 5 Parabacteroides merdae 6 19985 12637 1924 5424 0.8158308 63.2 9.6 27.1 6 Phocaeicola vulgatus 11 41005 12437 8376 20192 0.6463252 30.3 20.4 49.2

Using our metapangenome approach, we identified interesting patterns in accessory gene presence associated with antibiotic exposure (**Figure 7**). For example, the *Phocaeicola vulgatus* metapangenome is stable for the first 11 weeks of sampling even during ciprofloxacin exposure, but a portion of the accessory genome corresponding to XX% of the total metapangenome disappears at week 13, coinciding with metronidazole administration. While a portion of the *Parabacteroides merdae* metapangenome is present in early samples, the full genome is only detected after metronidazole administration when the fractional abundance of *P. merdae* increases starting at week 13. However, additional accessory elements are detected beginning at week 19, which coincides with a bloom of *P. merdae*.

In two bacterial species, antibiotic administration appears to spur on strain switching. In *Bacteroides uniformis*, one set of accessory elements present from weeks 0 - 11 is replaced by a new set in weeks 25-36. Similarly, in *Parabacteroides distasonis*, accessory elements present in weeks 4, 9, and 11 are replaced by new accessory elements in weeks 25-36. Both switches occur during metronidazole administration after the bloom of *P. merdae* and *P. vulgatus*.

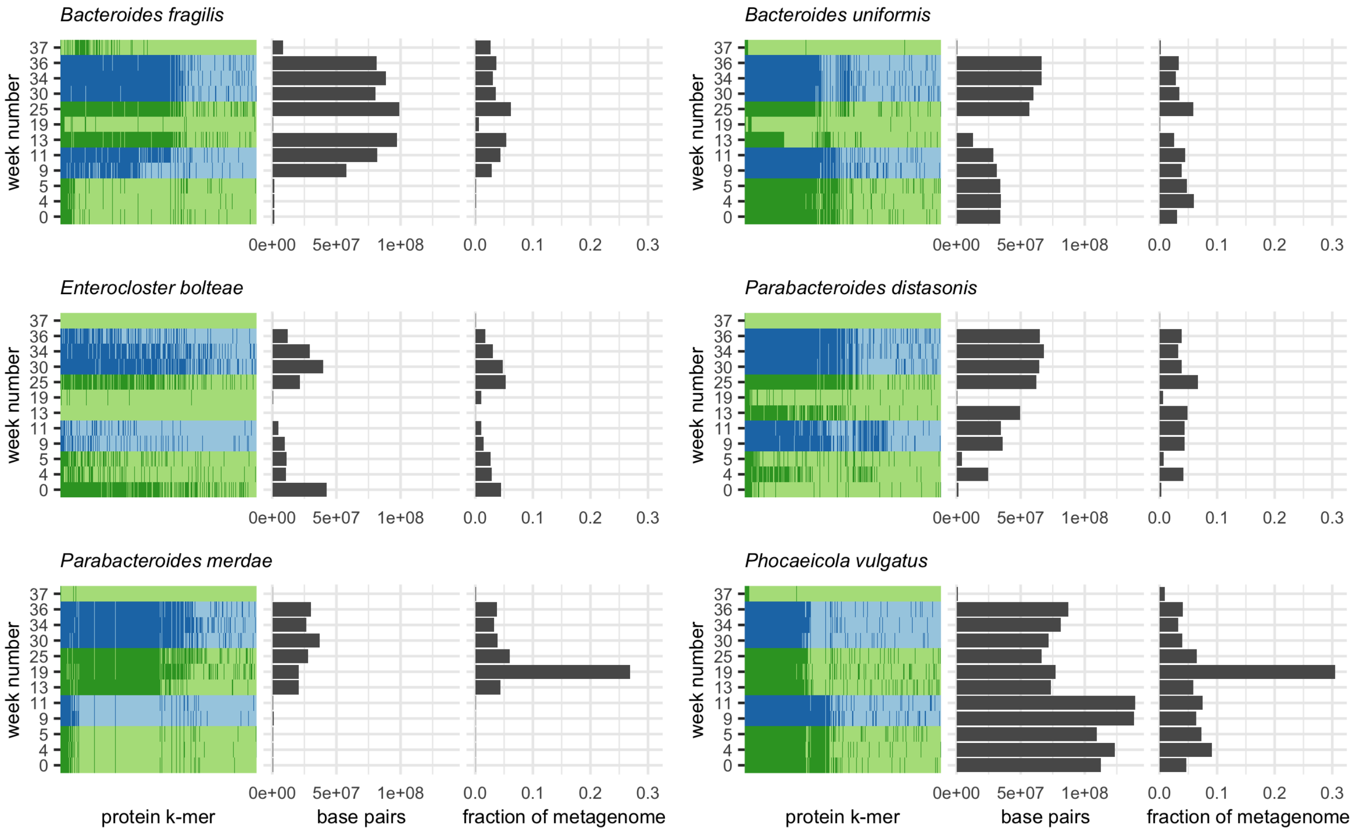

Figure 7:

- Do I need to compare these results against typical metapangenomics? like do de novo assembly, binning, prokka? etc?
- What else, if anything, belongs in this section?

Discussion

We present a method to perform assembly-free metapangenomics that is minimally reliant on reference databases. We show that pangenome metrics like core, cloud, and shell pangenome fractions can be accurately estimated with long amino acid k-mers and k-mers from other reduced alphabets. We then 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). 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 or map against reference databases, 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 metapangenome 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. Alternatively, species level databases were ~5 Mb in size, reducing the RAM and CPU tiem needed to run orpheum.

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 genomes 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.

These strategies may also be useful to predetermine the set of species-level databases to use for ORF prediction.

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.
- annotation is substantial drawback. Integrate potential of assembly graph annotation.

Methods

All code is available at github.com/taylorreiter/2021-panmers (results section 1), github.com/taylorreiter/2021-orpheum-sim (results section 2), and https://github.com/taylorreiter/2021-metapangenome-example (results section 3).

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.

Determining 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 :)

Metapangenome analysis of iHMP metagenomes

We used sourmash, spacegraphcats, and orpheum to peform metapangenome analysis of 12 iHMP time series gut microbiomes captured by short read shotgun metagenomes. We downloaded samples HSM6XRQB, HSM6XRQI, HSM6XRQK, HSM6XRQM, HSM6XRQO, HSM67VF9, HSM67VFD, HSM67VFJ, HSM7CYY7, HSM7CYYD, HSM7CYY9, HSM7CYYB from ibdmdb.org. We adapter and quality trimmed each sample with fastp (parameters `--detect_adapter_for_pe`, `--qualified_quality_phred 4`, `--length_required 31`, and `--correction`), removed human host sequencing reads with bbduk (parameters `k=31`, reference file https://drive.google.com/file/d/0B3llHR93L14wd0pSSnFULUlhcUk/edit?usp=sharing), and k-mer trimmed reads using khmer `trim-low-abund.py` (parameters `-C 3`, `-Z 18`, `-V`). We then used sourmash gather to infer the taxonomic profile of each sample, using the GTDB rs202 database (`k` = 31, https://osf.io/w4bcm/). We summarized the results to species-level using the GTDB taxonomy. We retained species with a cumulative sum of of at least 2% (sum of `f_unique_to_query`) across metagenome reads as query genomes. Within each species, we selected the genome with with largest cumulative `f_unique_to_query` across metagenomes. We downloaded each genome from GenBank (**Table 3**) and performed spacegraphcats assembly graph queries with each (parameters `ksize: 31`, `radius: 1`, `paired_reads: true`). Using the returned reads, we predicted open reading frames using orpheum `translate` (parameters `--jaccard-threshold 0.39`, `--alphabet protein`, `--peptide-ksize 10`) and using species-level GTDB databases. We sketched each set of translated reads using sourmash `sketch` (parameters `protein`, -p k=10,scaled=100,protein`), converted each sketch to a csv file, and then combined csv files for a single query species across all metagenomes.  This long format csv was used as input for the R pangenome package pagoo, using the `pagoo()`function. We used pagoo methods`pg$gg_binmap()‘, ‘pg$summary_stats()`, and`pg$pg_power_law_fit()` to visualize the pangenome, calculate the size of the core, shell, and cloud, and estimate alpha.

Table 3: Query genome GTDB species names and GenBank accessions.

species	accession
Parabacteroides distasonis	GCA_000162535.1
Enterocloster bolteae	GCF_003433765.1
Bacteroides fragilis	GCF_003458955.1
Parabacteroides merdae	GCF_003475305.1
Bacteroides uniformis	GCF_009020325.1
Phocaeicola vulgatus	GCF_009025805.1

References

1. **A new view of the tree of life**
Laura A Hug, Brett J Baker, Karthik Anantharaman, Christopher T Brown, Alexander J Probst, Cindy J Castelle, Cristina N Butterfield, Alex W Hernsdorf, Yuki Amano, Kotaro Ise, … Jillian F Banfield
Nature Microbiology (2016-04-11) https://doi.org/bpkh
DOI: 10.1038/nmicrobiol.2016.48 · PMID: 27572647

2. **A genomic catalog of Earth's microbiomes**
Stephen Nayfach, Simon Roux, Rekha Seshadri, Daniel Udwary, Neha Varghese, Frederik Schulz, Dongying Wu, David Paez-Espino, I-Min Chen, Marcel Huntemann, … IMG/M Data Consortium
Nature Biotechnology (2020-11-09) https://doi.org/ghjh4b
DOI: 10.1038/s41587-020-0718-6 · PMID: 33169036 · PMCID: PMC8041624

3. **Phage diversity, genomics and phylogeny**
Moïra B Dion, Frank Oechslin, Sylvain Moineau
Nature Reviews Microbiology (2020-02-03) https://doi.org/ggkq9f
DOI: 10.1038/s41579-019-0311-5 · PMID: 32015529

4. **Linking pangenomes and metagenomes: the <i>Prochlorococcus</i> metapangenome**
Tom O Delmont, AMurat Eren
PeerJ (2018-01-25) https://doi.org/gczf4x
DOI: 10.7717/peerj.4320 · PMID: 29423345 · PMCID: PMC5804319

5. **Global ecotypes in the ubiquitous marine clade SAR86**
Adrienne Hoarfrost, Stephen Nayfach, Joshua Ladau, Shibu Yooseph, Carol Arnosti, Chris L Dupont, Katherine S Pollard
The ISME Journal (2019-10-14) https://doi.org/gns4sb
DOI: 10.1038/s41396-019-0516-7 · PMID: 31611653 · PMCID: PMC6908720

6. **Meta-Pangenome: At the Crossroad of Pangenomics and Metagenomics**
Bing Ma, Michael France, Jacques Ravel
Springer Science and Business Media LLC (2020) https://doi.org/gns4r8
DOI: 10.1007/978-3-030-38281-0_9 · PMID: 32633911

7. **Metagenome-assembled genome binning methods with short reads disproportionately fail for plasmids and genomic Islands**
Finlay Maguire, Baofeng Jia, Kristen L Gray, Wing Yin Venus Lau, Robert G Beiko, Fiona SL Brinkman
Microbial Genomics (2020-10-01) https://doi.org/gns4sc
DOI: 10.1099/mgen.0.000436 · PMID: 33001022 · PMCID: PMC7660262

8. **Toward quantifying the adaptive role of bacterial pangenomes during environmental perturbations**
Roth E Conrad, Tomeu Viver, Juan F Gago, Janet K Hatt, Fanus Venter, Ramon Rosselló-Móra, Konstantinos T Konstantinidis
Cold Spring Harbor Laboratory (2021-03-15) https://doi.org/gns4sd
DOI: 10.1101/2021.03.15.435471

9. **MetaPalette: a <i>k</i> -mer Painting Approach for Metagenomic Taxonomic Profiling and Quantification of Novel Strain Variation**
David Koslicki, Daniel Falush
mSystems (2016-06-28) https://doi.org/gg3gbd

DOI: 10.1128/msystems.00020-16 · PMID: 27822531 · PMCID: PMC5069763

10. **Single-cell transcriptomics for the 99.9% of species without reference genomes**
Olga Borisovna Botvinnik, Venkata Naga Pranathi Vemuri, NTessa Pierce, Phoenix Aja Logan, Saba Nafees, Lekha Karanam, Kyle Joseph Travaglini, Camille Sophie Ezran, Lili Ren, Yanyi Juang, … CTitus Brown
Cold Spring Harbor Laboratory (2021-07-10) https://doi.org/gns4sg
DOI: 10.1101/2021.07.09.450799

11. **NCBI prokaryotic genome annotation pipeline**
Tatiana Tatusova, Michael DiCuccio, Azat Badretdin, Vyacheslav Chetvernin, Eric P Nawrocki, Leonid Zaslavsky, Alexandre Lomsadze, Kim D Pruitt, Mark Borodovsky, James Ostell
Nucleic Acids Research (2016-08-19) https://doi.org/f82gsk
DOI: 10.1093/nar/gkw569 · PMID: 27342282 · PMCID: PMC5001611

12. **RefSeq: expanding the Prokaryotic Genome Annotation Pipeline reach with protein family model curation**
Wenjun Li, Kathleen R O'Neill, Daniel H Haft, Michael DiCuccio, Vyacheslav Chetvernin, Azat Badretdin, George Coulouris, Farideh Chitsaz, Myra K Derbyshire, AScott Durkin, … Françoise Thibaud-Nissen
Nucleic Acids Research (2021-01-08) https://doi.org/gnrhsn
DOI: 10.1093/nar/gkaa1105 · PMID: 33270901 · PMCID: PMC7779008

13. **The Integrative Human Microbiome Project**
The Integrative HMP (iHMP) Research Network Consortium
Nature (2019-05-29) https://doi.org/gf3wp9
DOI: 10.1038/s41586-019-1238-8 · PMID: 31142853 · PMCID: PMC6784865

14. **Exploring neighborhoods in large metagenome assembly graphs using spacegraphcats reveals hidden sequence diversity**
CTitus Brown, Dominik Moritz, Michael P O'Brien, Felix Reidl, Taylor Reiter, Blair D Sullivan
Genome Biology (2020-07-06) https://doi.org/d4bb
DOI: 10.1186/s13059-020-02066-4 · PMID: 32631445 · PMCID: PMC7336657

15. **High throughput ANI analysis of 90K prokaryotic genomes reveals clear species boundaries**
Chirag Jain, Luis M Rodriguez-R, Adam M Phillippy, Konstantinos T Konstantinidis, Srinivas Aluru
Nature Communications (2018-11-30) https://doi.org/gfknmg
DOI: 10.1038/s41467-018-07641-9 · PMID: 30504855 · PMCID: PMC6269478

Appendix/Supplementary information

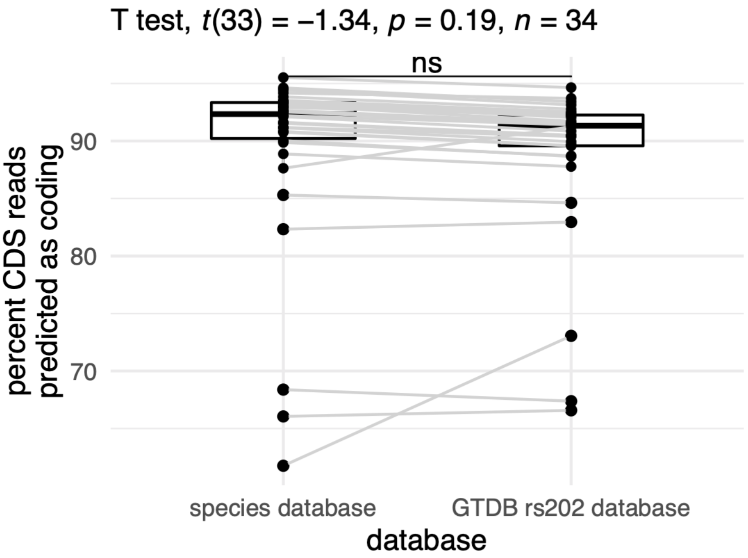

Figure 8: The slight increase observable for some species is a results in different thresholds, where we used 0.39 for the species database and 0.5 for the GTDB rs202 database.