

Structure and outline for today.

Morning lecture/lab:

- Introduction to shotgun metagenomics: The top 10 reasons why shotgun metagenomics is awesome!! - Titus
- K-mers; and Dude, what's in my metagenome? (part i) – Titus

Afternoon lecture/lab:

- Detecting and accounting for sample contamination – Ben
 - *mostly* 16S-focused

Evening open lab:

- Working through tutorials – R, 16S, low-level shotgun data analysis. (Ben, Titus, Maria et al.)

Evening social:

- Maybe a social game (werewolf)? ~8pm?

Introduction to Metagenomics

Titus Brown

MBL STAMPS 7/20/24

Hello and welcome!

- Your questions are a big part of the point of this lecture!
- Shotgun metagenomics is challenging, and even the most basic data understanding and analysis are hard.
- Today I am hoping to give you a foundation in what shotgun metagenomes are, and some intuition in how to think about their contents!
- This is a new lecture, specially crafted for STAMPS 2024 – *please* ask questions and provide feedback!!
- Annotated slides will be available after lecture.
- (Join me for lunch?)
- I am happy to do 1:1 or small-group whiteboarding/discussions on Sat or Mon evening on these topics, too!

I. The top 10 reasons why shotgun metagenomics is awesome!

1. Metagenomics samples everything!
2. Classifying genomes and metagenome content has become relatively straightforward and accurate.
3. Sequencing technology is improving fast!
4. Bioinformatics technology is improving quickly!
5. Metagenomics is differently biased from targeted approaches.
6. You can recover (dramatically) new genomes from metagenomes!
7. You can reuse metagenome data later in some very useful ways.
8. Metagenomics gets you down to strain level detection, in theory and maybe in practice.
9. It is theoretically and practically possible to do many *different* things with metagenomic data!
10. Shotgun metagenomicists are united by its difficulty!

1. Metagenomics samples everything!

- Within the limits of sample diversity and molecule “type” (DNA by default), shotgun metagenomics samples everything.
- This includes viruses, bacteria, archaea, and eukaryotes...
- ...as well as “garbage”
 - Degraded DNA/RNA
 - Contaminants and unintended companions
- ...as long as you sequence deeply enough.

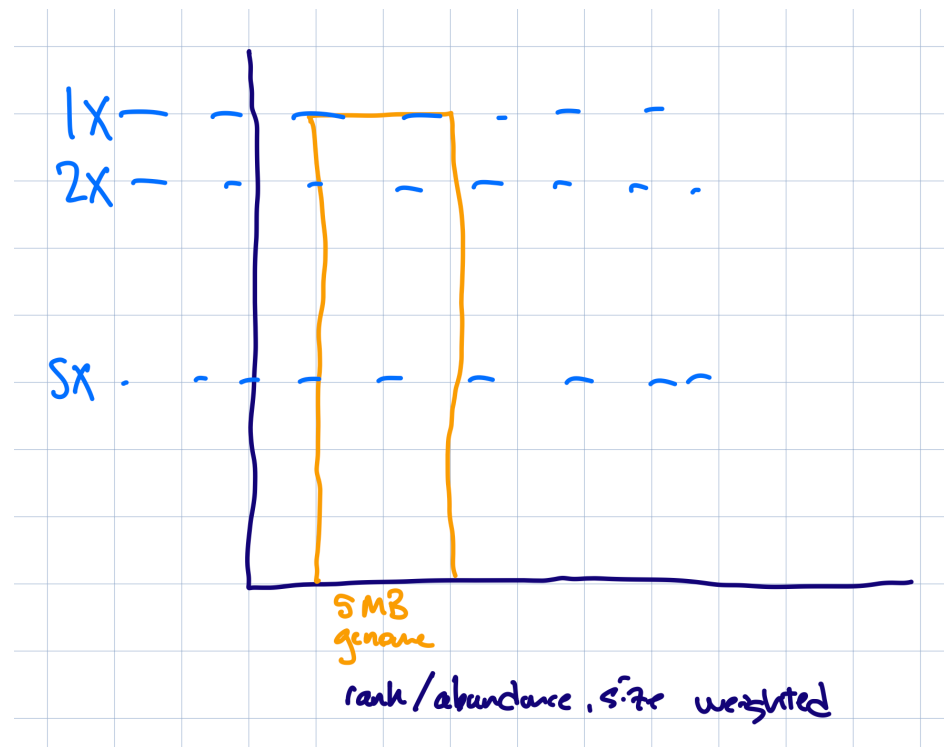
Why is 16S so much deeper than shotgun metagenomics for community sampling??

Consider a mixture of 1000 genomes, each 1 MB in size, all equimolar.

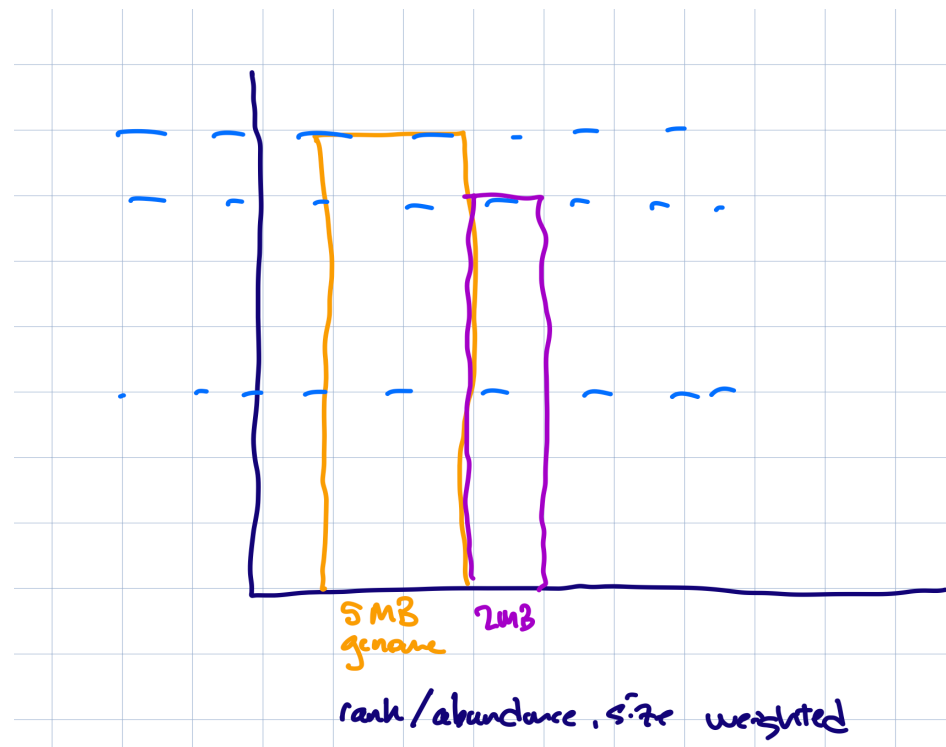
Sequencing all 1000 genomes to 1x coverage would require a billion base pairs (1000 genomes x 1 million bases).

Instead, if you can select a 1 kb region as “diagnostic” of genome presence, you can sequence just a million base pairs to get that same answer.

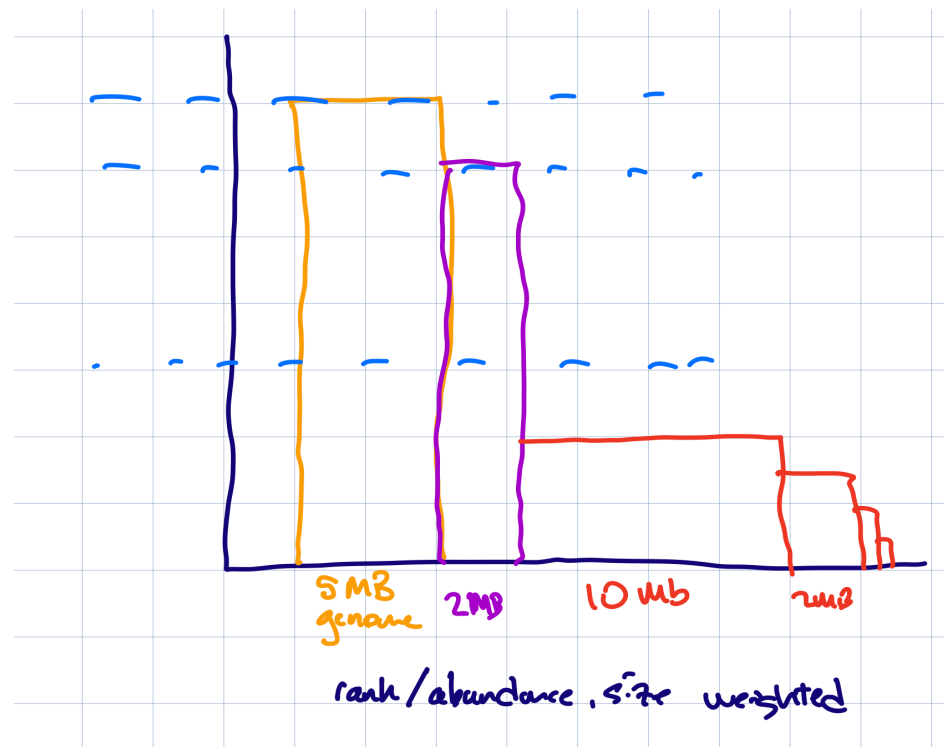
Sequencing depth and sample diversity (1)



Sequencing depth and sample diversity (2)



Sequencing depth and sample diversity (3)



How deeply should I sequence??

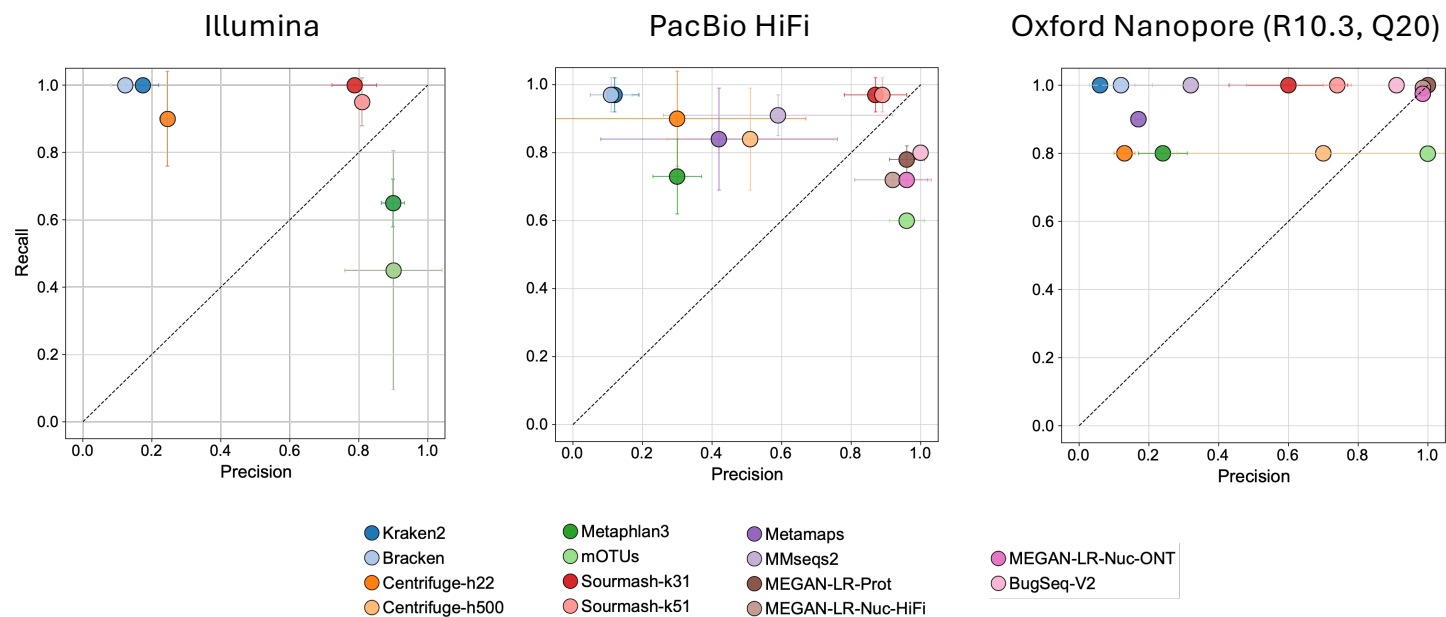
- The *measure* is “bp generated”. But:
- There is no straightforward empirical answer:
 - Unknown contents (including things that 16S can’t detect).
 - Sample and sequencing variability.
 - Strain level heterogeneity.
- The best practical answer is:
 - Target 2x what other people have used when working on similar samples.
 - Mix and match approaches and technology, e.g.
 - 16S for *many* samples, shotgun metagenomics for a few samples to provide references.
 - Short reads for sample characterization, long reads for genome recovery.
- I might suggest an 80/20 perspective: focus 80% of your effort on addressing your core questions, and 20% on cool new stuff.
- Remember that *generating* data is just the beginning: you also need to be able to analyze it and reach robust conclusions...

2. Classifying genomes and metagenome content has become relatively straightforward and accurate.

A few big changes in recent years:

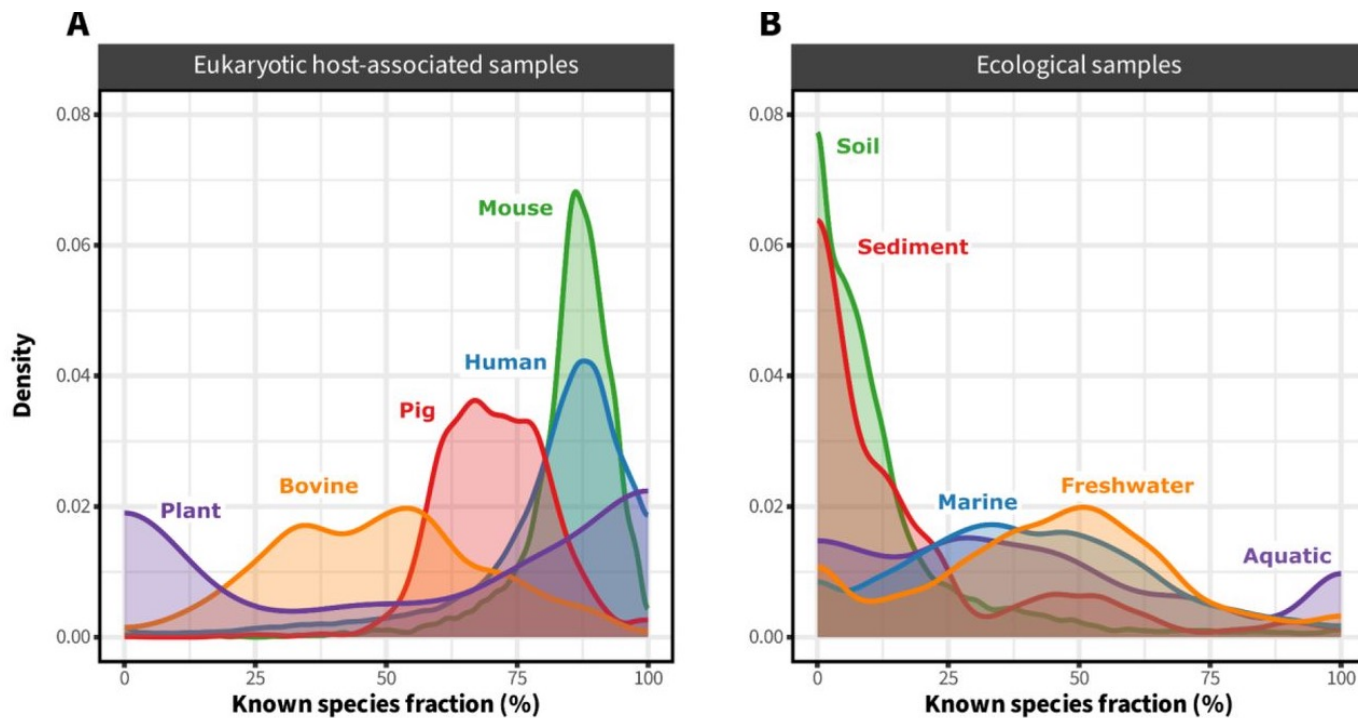
- The GTDB taxonomy (for bacteria and archaea) is great!
 - Species-level classifications are based on genome sequence, unlike NCBI; this makes it possible for sequence-based analysis tools to perform well.
 - Higher-level (class, order, etc.) classifications are based on ribosomal proteins, and tools to do that well are also readily available and becoming faster (GTDB-Tk, SingleM, FastAAI).
- Large numbers of genomes are increasingly available (~3m) which makes it possible to do ~perfect species level classification.
- Metagenomic classification works increasingly well.

There are sensitive and specific options for taxonomic classification of metagenomes.



Portik et al., 2022

The landscape of metagenome classification



(Sandpiper & singleM are fantastic!!)

doi: 10.1101/2024.01.30.578060

Remaining frontiers in taxonomy -

- We need better characterization of genomes relevant to environmental samples! (But this will come.)
- GTDB only covers bacteria and archaea...
 - Viral taxonomy is really hard, because viruses are incredibly variable; this area is undergoing rapid development!
 - Eukaryotic taxonomy needs development as well. (I know very little about the challenges here.)
- Strain level classification is important, challenging and (potentially) non-sensical due to genome mixing.

3. Sequencing technology is improving fast!

- Ben covered this really well, but PacBio HiFi gives you basically “perfect” genomes, and sequencing costs are dropping.
- If you can use short-read sequencing (e.g. for taxonomic and functional profiling), then life is pretty good, I think.
- But, remaining challenges combine synergistically ☹
 - The cost of HiFi is high.
 - Sample concentration required is unreachable for some samples.
 - Metagenomic complexity (diversity & richness) for some samples (soil, sediment, maybe water) is hundreds to thousands of times higher than (e.g.) gut.
- Multiple strains challenge even long-read assembly.
 - See Feng & Li, 2024, doi: 10.1186/s13059-024-03234-6
 - Will be discussed on Monday by Todd et al.

4. Bioinformatics technology is improving quickly!

- There is a profusion of tools to do your initial data analyses, whatever they may be!
- Conda has made it straightforward to install most command-line tools.

That having been said,

- Sorting out which programs to use is definitely a challenge 😓.
- My recommendation is to start your journey with a tool that
 1. Has worked for someone else, preferably in similar circumstances.
 2. Executes to completion.
 3. Gives you useful output.
 4. Has decent documentation.

5. Metagenomics is differently biased

- Targeted approaches (typically based on PCR or array capture) typically rely on ultraconserved regions.
- This privileges what is *already known*.
- In particular, “primer bias” can lead to emphasizing measurement of what we already know pretty well.
- 16S also falls afoul of copy number variability: the number of (identical) 16S regions in a bacterial genome can vary ~10 fold within a species.
- Shotgun metagenomics is not *unbiased* by any means, but has a *different* bias - typically related to GC content in DNA, as well as other details of the sample processing specific to shotgun sequencing.

Some reading!

Wavy coverage patterns in mapping results – A. Murat Eren (Meren). [link](#)

Consistent and correctable bias in metagenomic sequencing experiments, McLaren, Willis, and Callahan, 2019.

- tl;dr a really nice summary of sources of bias in metagenomic experiments.

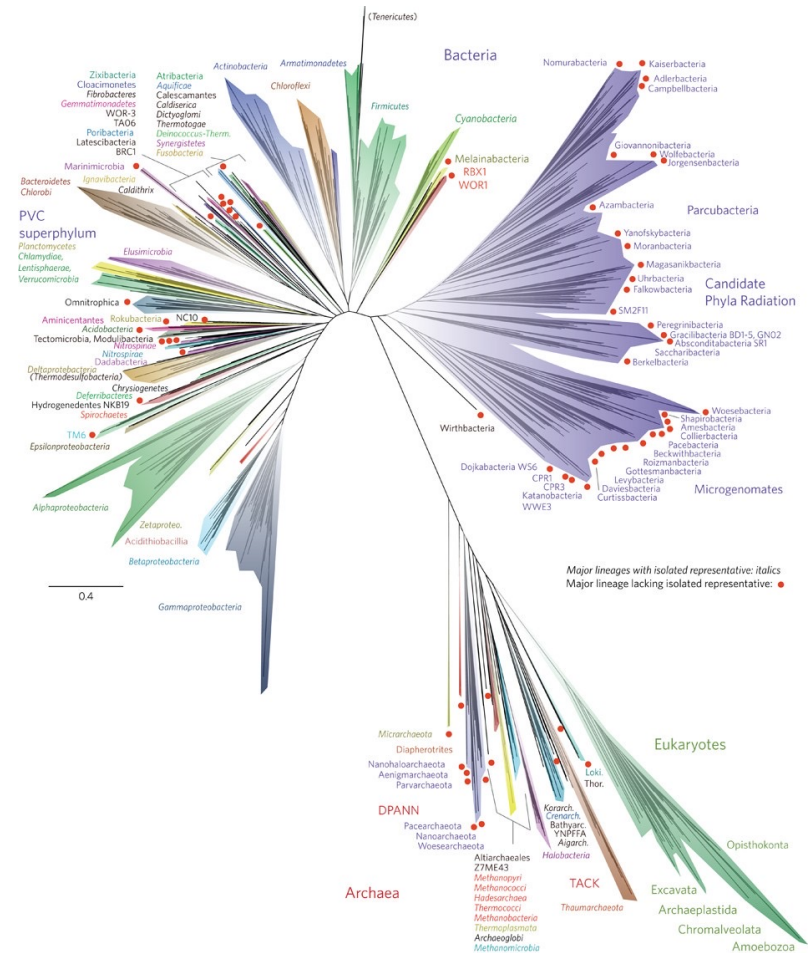
Nitrogen-fixing populations of Planctomycetes and Proteobacteria are abundant in surface ocean metagenomes, Delmont et al., 2018.

- tl;dr amplicon-based strategies of measurement systematically underestimated the abundance of diazotrophic heterotrophs in the ocean for decades.

6. You can recover (dramatically) new genomes from metagenomes!

Metagenome-assembled genomics (and other related methods) have been pulling an increasing number of Very Different bacterial, archaeal, viral, and even eukaryotic genomes from metagenome samples.

(Methods will be discussed on Monday!)

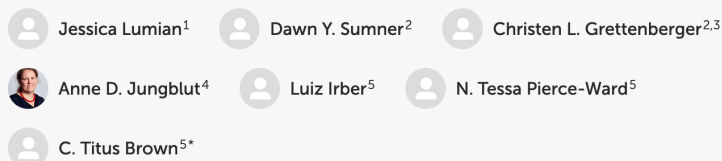


Hug et al., 2016. doi: 10.1038/nmicrobiol.2016.48

7. You can reuse metagenome data later in some very useful ways.


- Publicly available metagenomes remain very useful for a variety of *new* analyses years after they are published!

Biogeographic distribution of five Antarctic cyanobacteria using large-scale k-mer searching with sourmash branchwater



doi: [10.3389/fmicb.2024.1328083](https://doi.org/10.3389/fmicb.2024.1328083)

Petabase-scale sequence alignment catalyses viral discovery

[Robert C. Edgar](#), [Brie Taylor](#), [Victor Lin](#), [Tomer Altman](#), [Pierre Barbera](#), [Dmitry Meleshko](#), [Dan Lohr](#), [Gherman Novakovsky](#), [Benjamin Buchfink](#), [Basem Al-Shayeb](#), [Jillian F. Banfield](#), [Marcos de la Peña](#), [Anton Korobeynikov](#), [Rayan Chikhi](#) & [Artem Babaian](#) 

doi: [10.1038/s41586-021-04332-2](https://doi.org/10.1038/s41586-021-04332-2)

(demo: <https://branchwater.jgi.doe.gov/>)

8. Metagenomics gets you down to strain level detection, in theory and maybe in practice.

- 16S can resolve genera and some species, but not below.
- Shotgun metagenomics can sensitively determine presence/absence and abundance of specific *genomes*, and hence can resolve *strains*.
- From a bioinformatics perspective, this is a ~solved problem, I think.
- Biology disagrees ☹
 - The genomic content of metagenomes is rarely a perfect match to your reference sequence!
 - Genomes seem to remix and recombine at the species level.
- Also: *recovering* new strain-level genomes from metagenomes is hard.
- Both Todd and I will discuss all of this more (and maybe disagree!) on Monday.

9. It is theoretically and practically possible to do many *different* things with metagenomic data!

- Community diversity/richness.
- Analysis of known bac/arc/viral/euk presence/absence.
- Abundance of genomes.
- Detection of core/accessory elements (pangenomics).
- Analysis of functional content.
- Analysis of Antimicrobial Resistance Genes.
- Discovery of novel proteins and pathways (e.g. biosynthetic gene clusters).

Corollary: Shotgun metagenomics offers many opportunities for bioinformatics “growth” 😄😭.

10. Shotgun metagenomicists are united by its difficulty!

- Methods developers for metagenomics are (in my experience) mostly just thankful to get a result that is not wildly wrong, and are friendly and helpful (perhaps as a result?)
- There are a bunch of rich, powerful techniques that we can mix and match to get answers!
 - Mapping
 - Assembly, binning
 - K-mers
- ...but applying them appropriately remains an ongoing challenge.
- **Scale** (10s of millions of genomes, millions of samples, extremely large sequence data sets) is an ongoing challenge as well.

II. Dude, what's in my metagenome? (part i)

Interrogating metagenomes through k-mers for fun and nonprofit!

K-mer based techniques provide *sensitive* and *specific* ways to analyze “raw” genome and shotgun metagenome content.

These are really low-level and granular analyses; today is mostly about building intuition and understanding, not about high-level characterization!

What are k-mers?

Fixed-length “words” of DNA that are extracted by sliding a window along sequence.
“k” is the window size.

```
[12]: build_kmers('ATGGACCAGATATAGGGAGAGCCAGGTAGGACA', 21)
```

```
[12]: ['ATGGACCAGATATAGGGAGAG',  
      'TGGACCAGATATAGGGAGAGC',  
      'GGACCAGATATAGGGAGAGCC',  
      'GACCAGATATAGGGAGAGCCA',  
      'ACCAGATATAGGGAGAGCCAG',  
      'CCAGATATAGGGAGAGCCAGG',  
      'CAGATATAGGGAGAGCCAGGT',  
      'AGATATAGGGAGAGCCAGGTA',  
      'GATATAGGGAGAGCCAGGTAG',  
      'ATATAGGGAGAGCCAGGTAGG',  
      'TATAGGGAGAGCCAGGTAGGA',  
      'ATAGGGAGAGCCAGGTAGGAC',  
      'TAGGGAGAGCCAGGTAGGACA']
```

What are k-mers? Features of note:

- Can be generated from *any* sequence – in particular, both assembled genomes/metagenomes, and unassembled data sets.
 - (We will talk more about assembly on Monday!)
- K-mer matching between samples is based on identity – 100% match.
- DNA k-mers are typically "collapsed" around reverse-complement matches (e.g. we regard ATGCCC as equivalent to GGGCAT).
- We often use odd k-mer sizes to avoid DNA palindromes.
- You can also extract k-mers from protein sequences, but for today we will focus on DNA k-mers.

K-mers are really sensitive *and* really specific (in theory)

There are 4,398,046,511,104 possible k-mers of size 21 ($1 / 4^{**21}$)

You would expect approximately 5 k-mers to be shared between two unrelated genomes, each of size 5 MB.

There are 4,611,686,018,427,387,904 possible k-mers of size 31.

$E = 5 \times 10^{-6}$ for two k-mers to be shared.

Mathematically, it is easy to make k-mers really specific!

And, for perfect sequence, you can “call” a match based on a single observation => sensitivity.

But does biology agree?

K-mers are really sensitive *and* can be really specific (in practice)

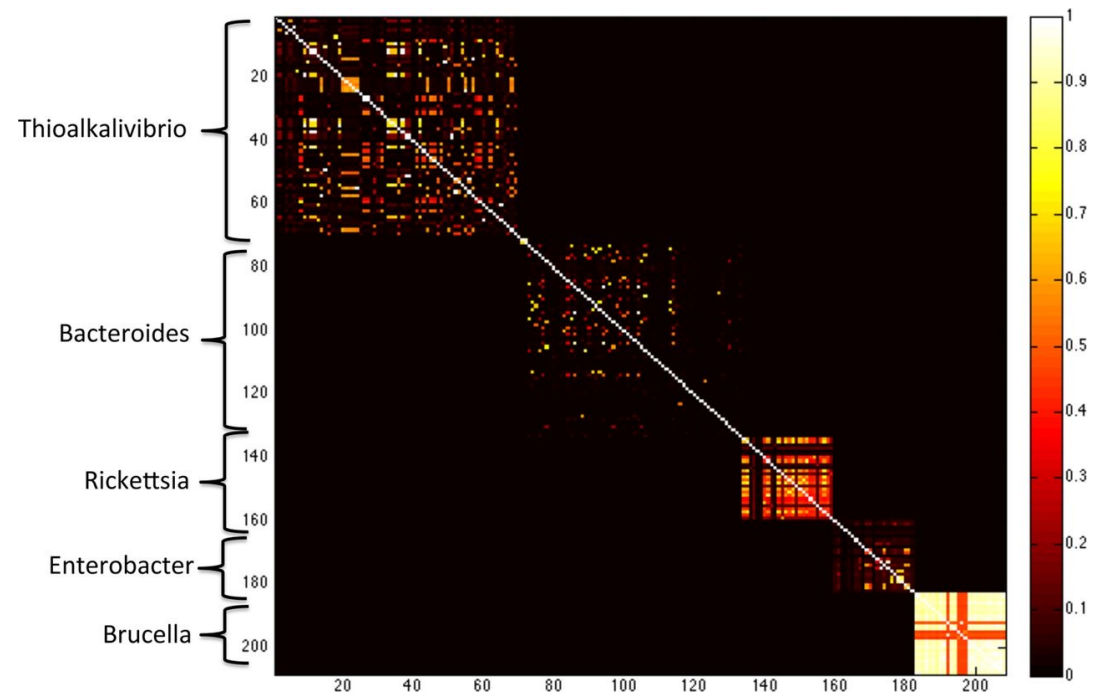
K-mers are very species specific when they are long enough!

(Here: $k=40$.)

$K=21$ – genus

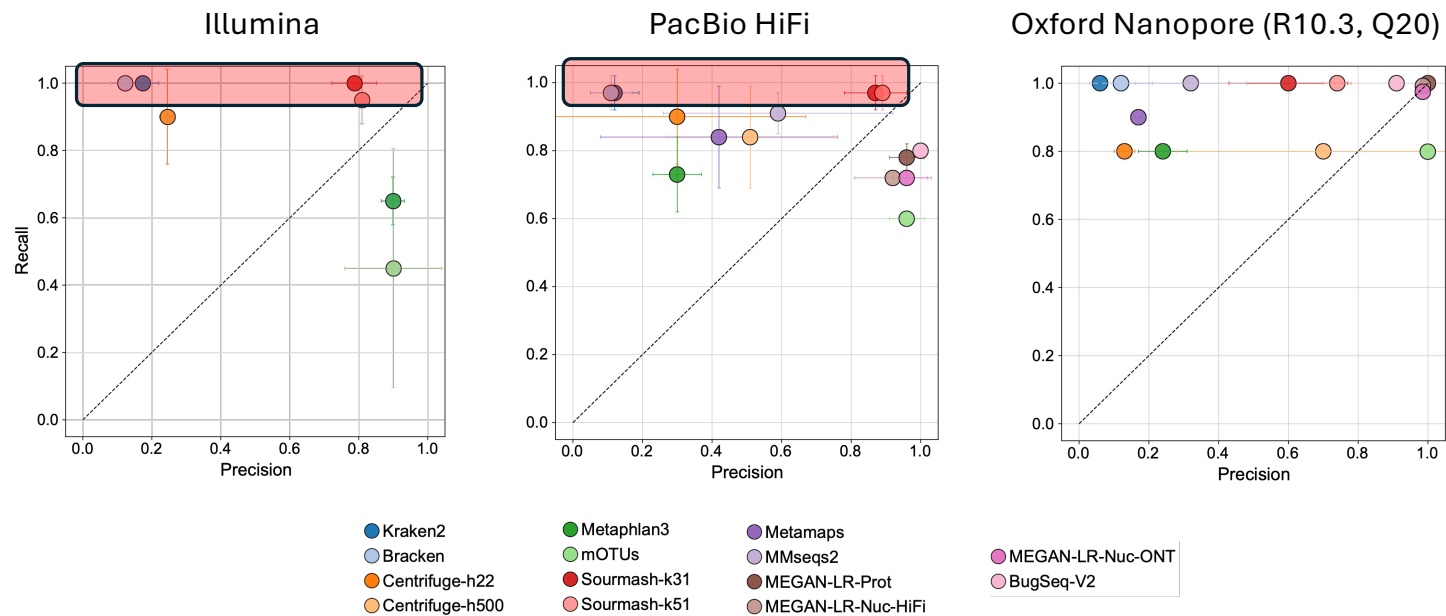
$K=31$ – species

$K=51$ – strain



Koslicki and Falush, doi: 10.1128/msystems.00020-16

Notice! The highest-recall methods for taxonomic classification are k-mer based!



Portik et al., 2022

Computers (and computer scientists) like k-mers!

- One parameter: k .
- Exact matches are easy for everyone to understand!
 - You can convert k-mers into numbers via *hashing* and then just compare numbers. Computers like numbers.
- Really efficient techniques for large-scale use via *sketching*.
 - We will discuss this more on Monday.

Two downsides of k-mers

There are *lots* of them and sequencing errors *make more*.

```
[12]: build_kmers('ATGGACCAGATATAGGGAGAGCCAGGTAGGACA', 21)
```

```
[12]: ['ATGGACCAGATATAGGGAGAG',  
      'TGGACCAGATATAGGGAGAGC',  
      'GGACCAGATATAGGGAGAGCC',  
      'GACCAGATATAGGGAGAGCCA',  
      'ACCAGATATAGGGAGAGCCAG',  
      'CCAGATATAGGGAGAGCCAGG',  
      'CAGATATAGGGAGAGCCAGGT',  
      'AGATATAGGGAGAGCCAGGTA',  
      'GATATAGGGAGAGCCAGGTAG',  
      'ATATAGGGAGAGCCAGGTAGG',  
      'TATAGGGAGAGCCAGGTAGGA',  
      'ATAGGGAGAGCCAGGTAGGAC',  
      'TAGGGAGAGCCAGGTAGGACA']
```


How do you compare collections of sequences using k-mers??

Set operations!

```
[1]: def jaccard_similarity(a, b):  
    a = set(a)  
    b = set(b)  
  
    intersection = len(a.intersection(b))  
    union = len(a.union(b))  
  
    return intersection / union
```

```
[2]: def jaccard_containment(a, b):  
    a = set(a)  
    b = set(b)  
  
    intersection = len(a.intersection(b))  
  
    return intersection / len(a)
```

Note: Jaccard is a distance metric; containment is not.

```
[3]: a = ['ATGG', 'AACC']  
     b = ['ATGG', 'CACA']  
     c = ['ATGC', 'CACA']
```

```
[4]: jaccard_similarity(a, a)
```

```
[4]: 1.0
```

```
[5]: jaccard_containment(a, a)
```

```
[5]: 1.0
```

```
[6]: jaccard_similarity(b, a)
```

```
[6]: 0.3333333333333333
```

```
[7]: jaccard_similarity(a, c)
```

```
[7]: 0.0
```

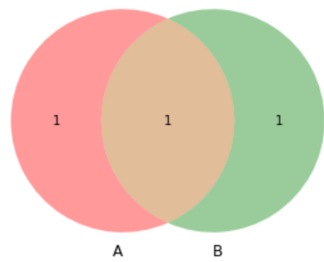
```
[8]: jaccard_containment(b, a)
```

```
[8]: 0.5
```

```
[3]: a = ['ATGG', 'AACC']  
     b = ['ATGG', 'CACA']  
     c = ['ATGC', 'CACA']
```

```
venn2([set(a), set(b)])
```

```
[9]: <matplotlib_venn._common.VennDiagram at 0x132a99340>
```



```
[10]: venn3([set(a), set(b), set(c)])
```

```
[10]: <matplotlib_venn._common.VennDiagram at 0x132c42430>
```



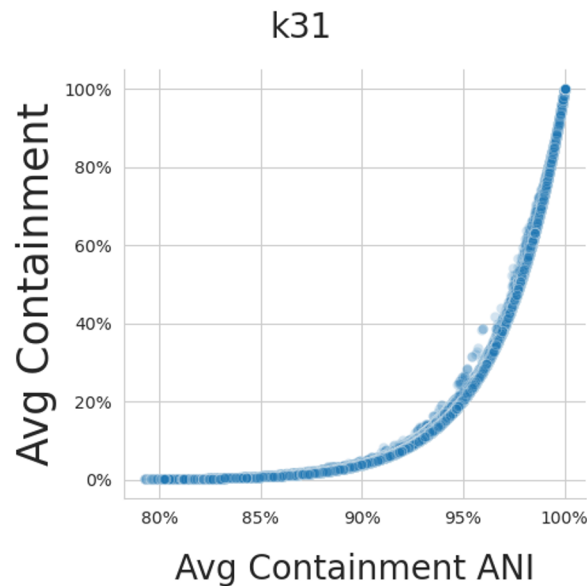
Digression: “Average Nucleotide Identity” vs k-mers

ANI is an alignment based measure of sequence similarity:

- align two sequences (line them up with the smallest number of mismatches possible)
- calculate the fraction of bases that are identical
- ~95% is considered “same species”

ANI has been shown to correlate well with hybridization-based measures of DNA similarity (“COT curves”)

K-mer containment can be converted to/from Average Nucleotide Identity – they are log related.



Estimating the average containment to ANI relationship for k = 31

K-mer containment is something we can calculate with our tool quickly and easily from *unassembled* sequence data.

Average Nucleotide Identity can usually only be calculated from assemblies.

But we can convert between alignment free and alignment-based measures!

Work by Tessa Pierce-Ward, David Koslicki, and Mahmud Rahman.
Rahman Hera et al., 2022,
<https://doi.org/10.1101/2022.01.11.475870>

Alignment-based vs k-mer-based (alignment-free)

- tl;dr; They are interconvertible within a useful range!
- ANI has trouble separating sequences with a very high level of similarity (99.9%, 99.99%, 99.999% ...)
- K-mers cannot robustly detect dissimilar sequences (< 90% ANI).
- Use whatever technique works for your question of interest.

Increasingly, tool developers are mixing and matching k-mers and alignment based techniques behind the scenes anyway.

The *biggest* difference is that you *do not need to assemble datasets* to use k-mers 😊😈

Multiplicity and k-mer abundance

- The above measures (Jaccard, containment) are all “flat” - they operate on presence/absence of k-mers.
- But for many situations (especially metagenomes!) you care *how many times* a sequence is present.
- For k-mers, the equivalent is “multiplicity”.
- This is a less-well-developed area, bioinformatically.
- I refer to this as “abundance”, below.
- (More this evening, or on Monday.)

Let's compare some genomes and metagenomes!!!

These kinds of comparisons underlie many actual genomic metagenomic analyses:

- Distance comparisons
- Overlap/matching of content
- Compositional analysis

...but are usually presented in summary form.

We'll start at “base” of conceptual hierarchy of analyses, and try to connect to usual abstractions! Wish us luck!

We're going to play a game!

- I give you a diagram or picture with some details.
- You think/pair/share:
 - 1 minute of looking at diagram alone;
 - 2-4 minutes of discussion and questions;
- We discuss as a class!
 - Ask questions
 - Uncover assumptions
 - Delineate what can and can't be inferred
- (Note: You will be able to run these analyses yourself this evening!)
- (Note 2: some of these analyses break my brain.)

Diagram 0!

Here is a comparison of k-mers from three genomes, and their overlap!

k=31, DNA. No abundance.

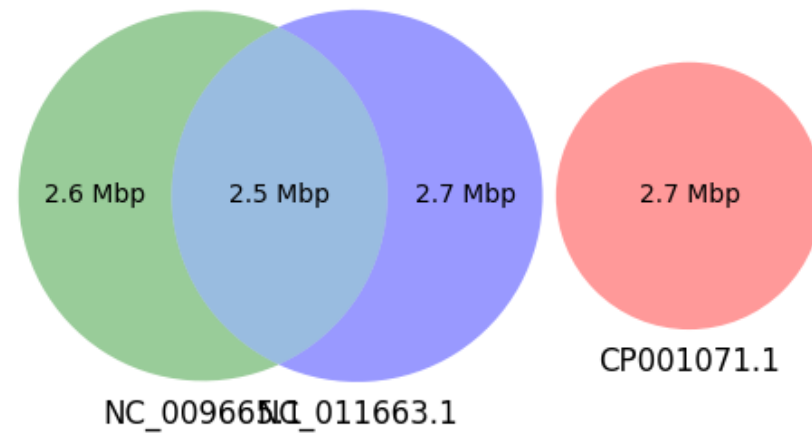


Diagram 0: Jaccard similarity matrix.

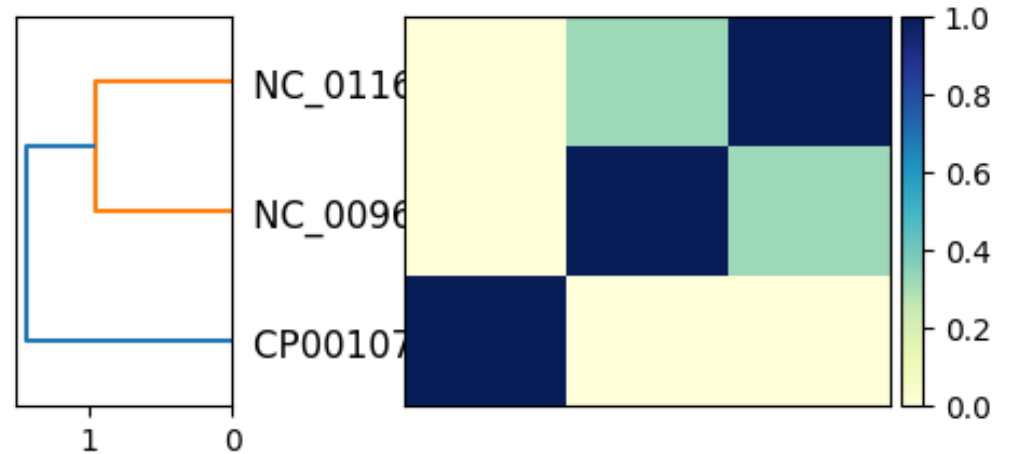
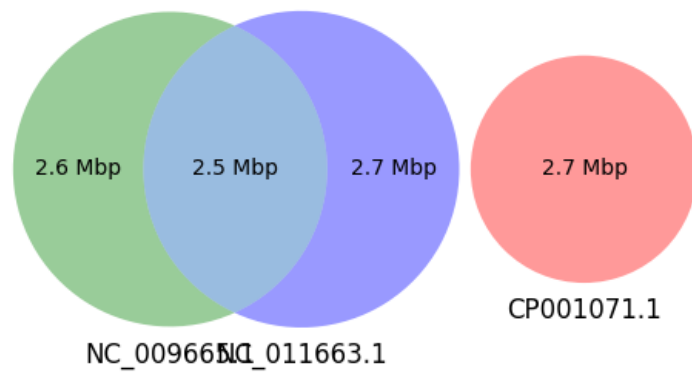
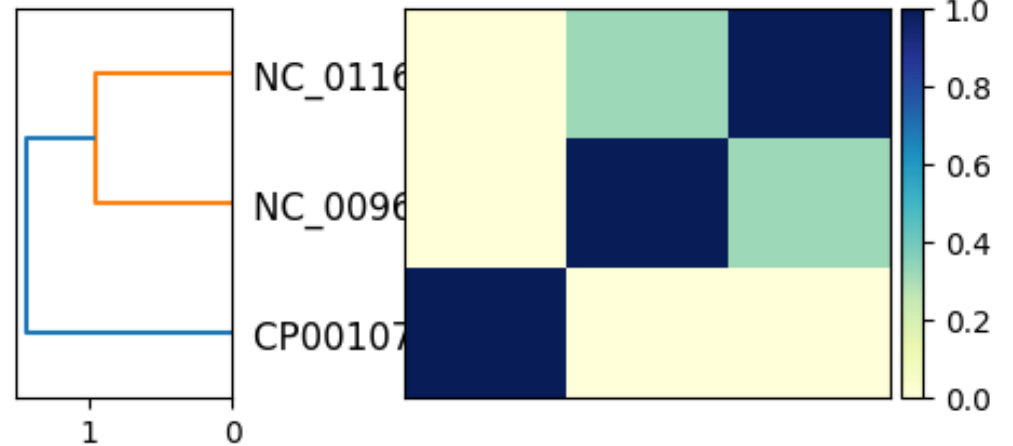
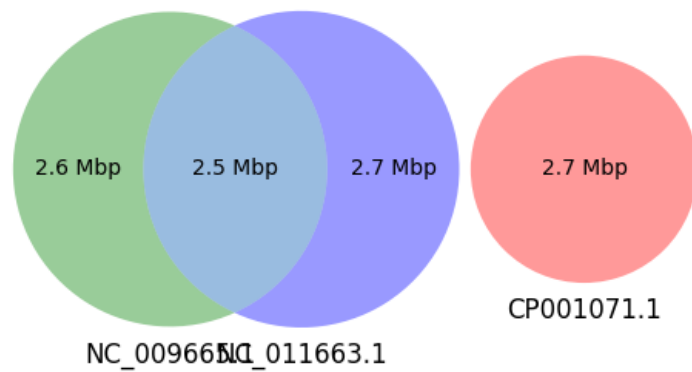


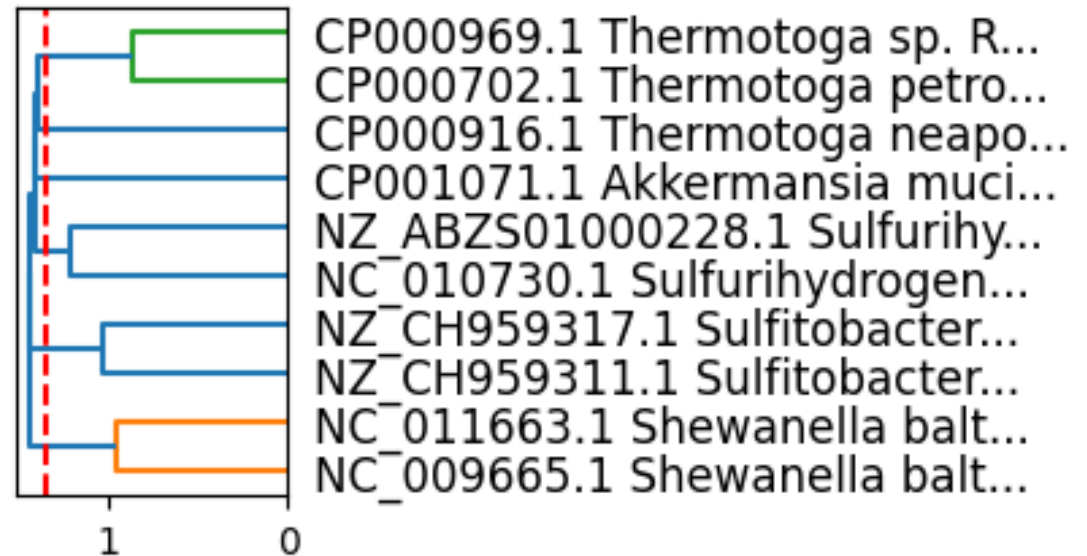
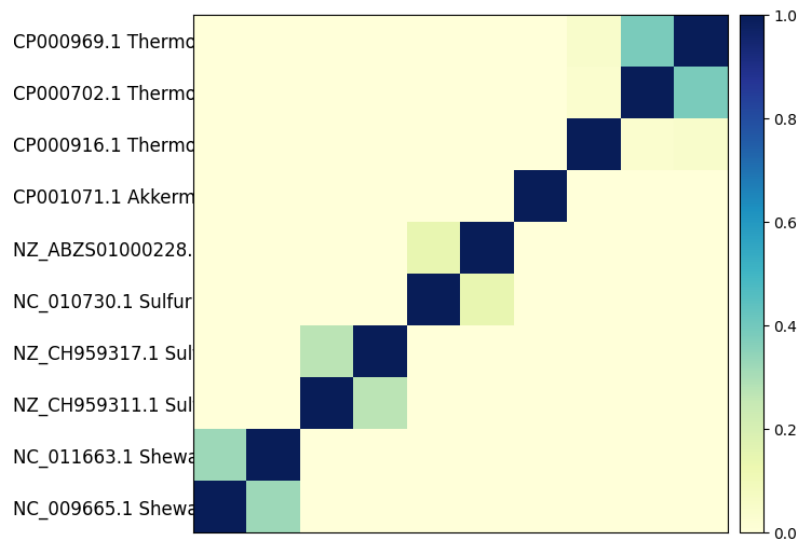
Diagram 0: Jaccard similarity



Matrix to dendrogram for 10 samples (Jaccard)

Scikit-learn “linkage” dendrogram.

This is essentially a neighbor joining tree.



Distance matrices \Leftrightarrow dendrograms \Leftrightarrow ordination plots

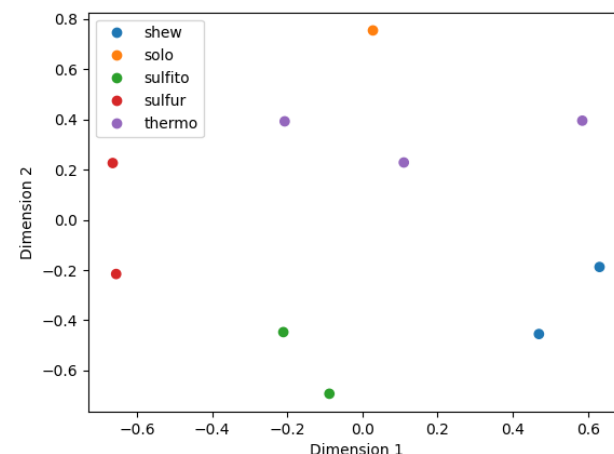
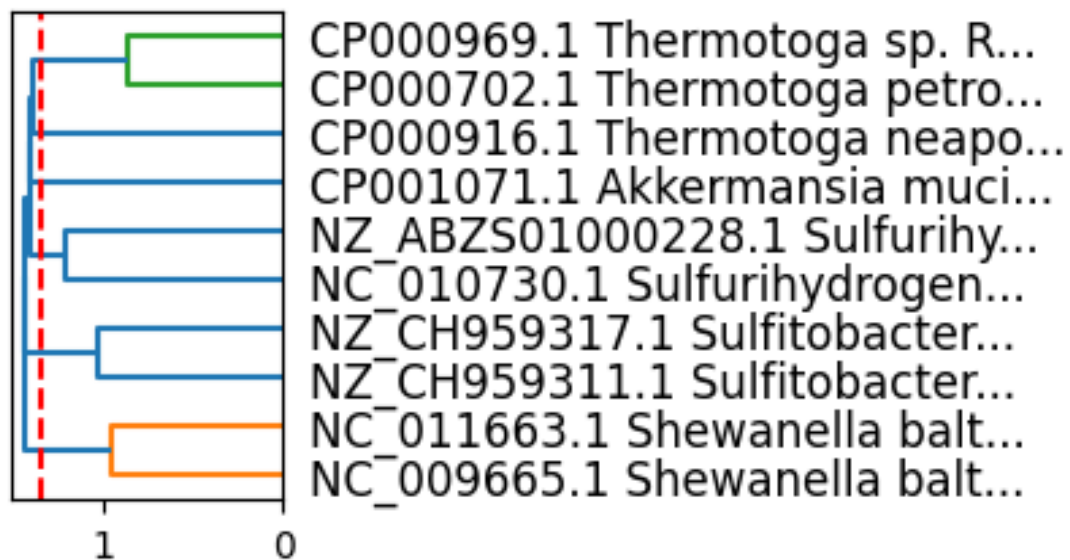


Diagram 1: Metagenome comparison.

K=31, DNA. No abundance.

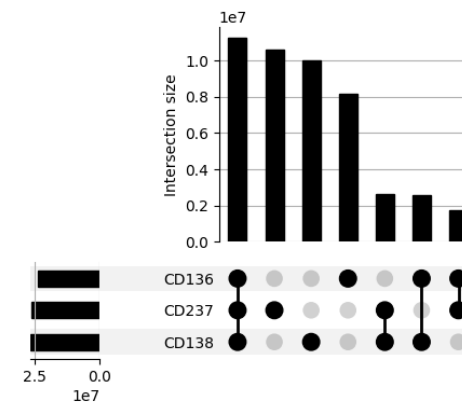
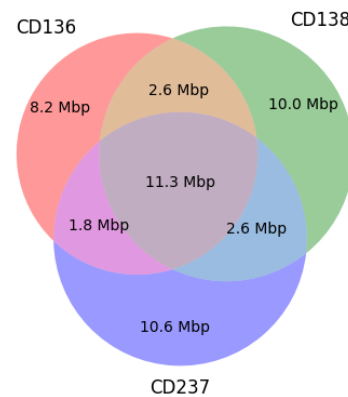


Diagram 2: Genomes and metagenomes

K=31, DNA. No abundance.

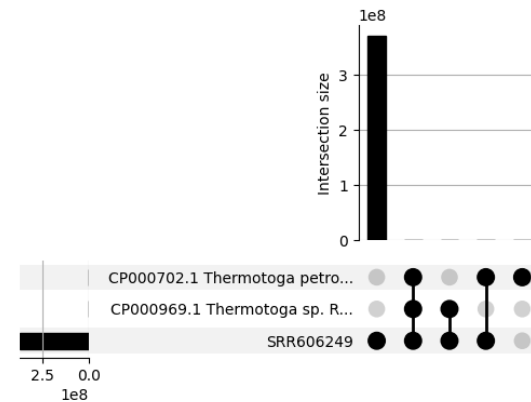
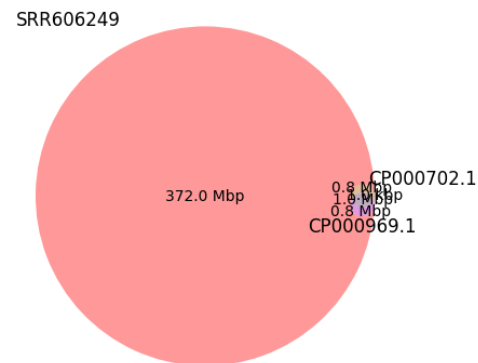


Diagram 3: Abundance histograms of k-mers in metagenomes

K=31, DNA.

With abundance.

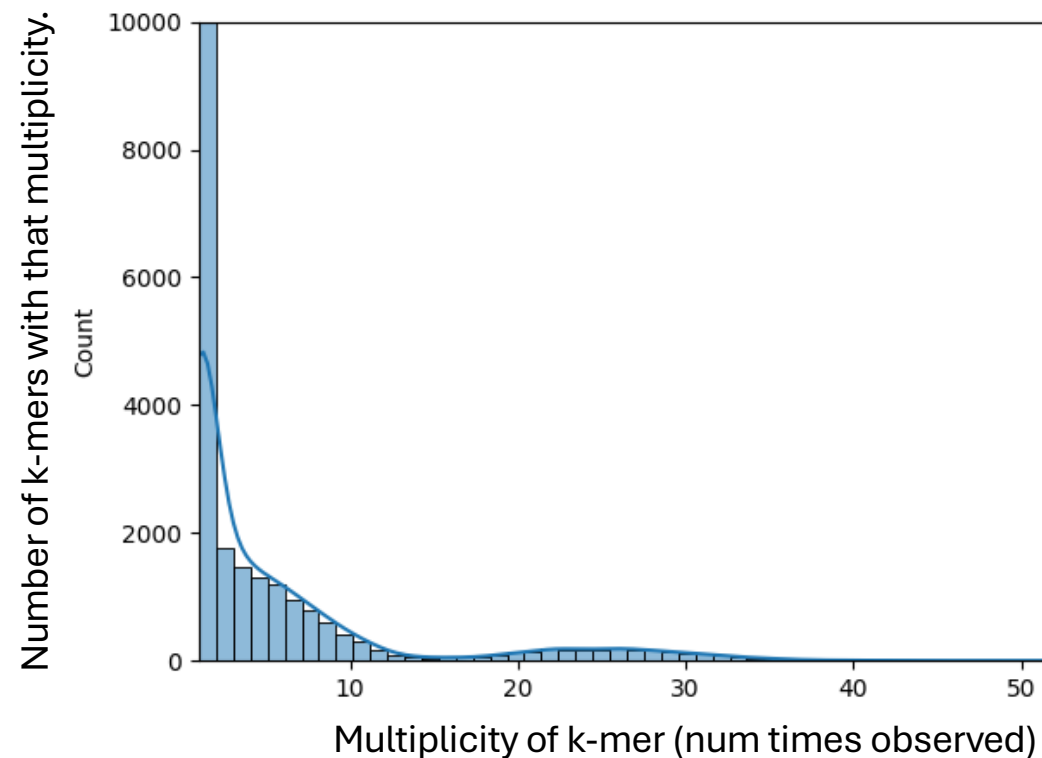


Diagram 4: Presence/absence content plots

K=31, DNA.
10kb regions from
known *B. fragilis*
genomes.
No abund.

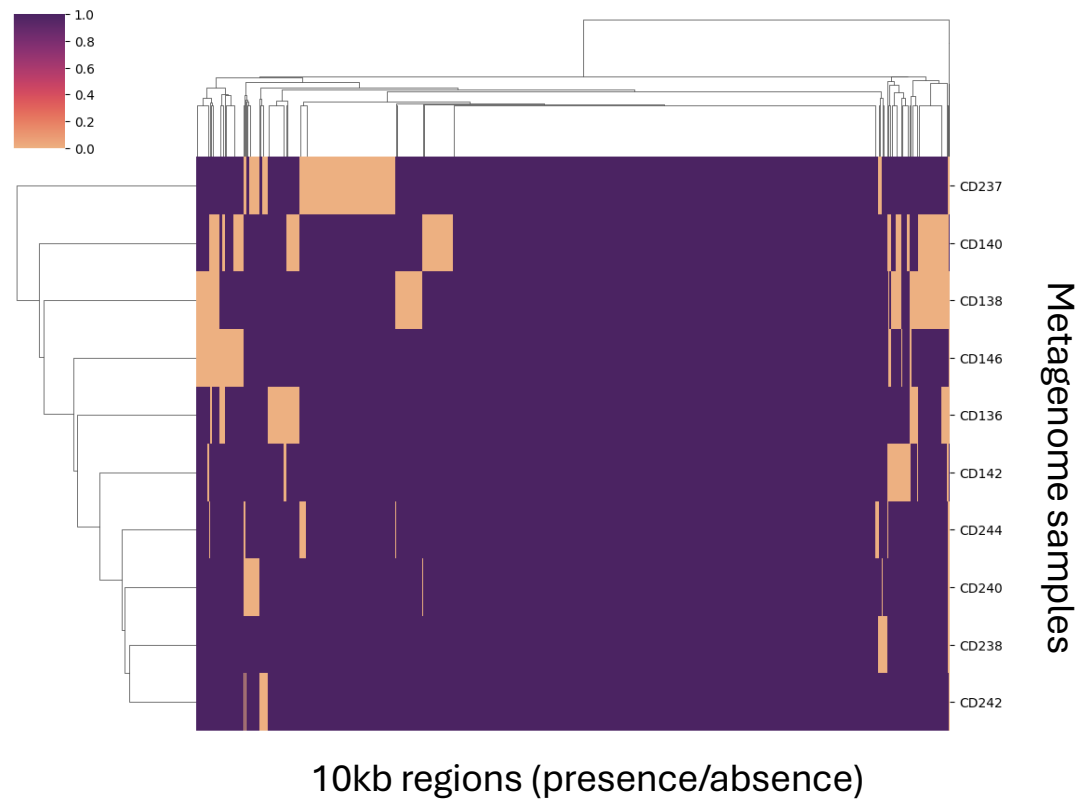


Diagram 5: Which genomes are present in this metagenome?

K=21, DNA. Abund.

All points are robustly observed under a naïve null model.

