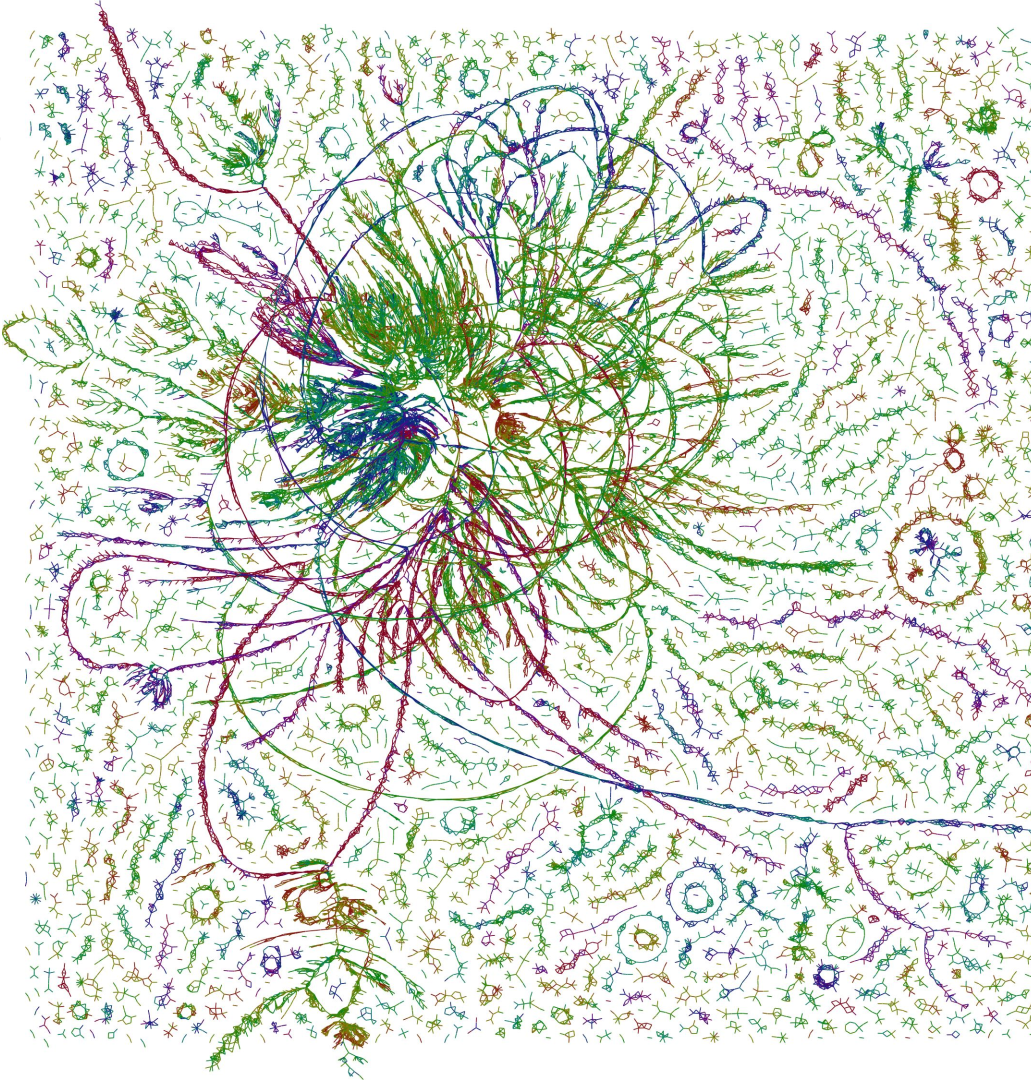
Metagenomics Lesson 3

Metagenomics Lesson 3

Taxonomic Classification using MinHash

1. MinHash algorithm
2. Demo: Classification using sourmash
3. Demo: Quality trimming using bbduk
4. Demo: Classification using sendsketch



Set Similarity: Jaccard index

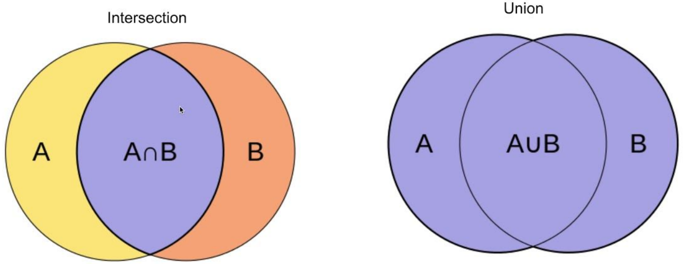
Let’s say we have two genomes that we’d like to compare based on their gene content.

In this case a *set* is composed of a collection of genes in each genome.

The Jaccard Similarity is a measure of set similarity given by the intersection of the sets divided by their union.

That is, we count the number of shared genes (encoded in both genomes), and divide that by the total number of unique genes that both genomes collectively encode.

Jaccard coefficient



$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

If Genome A has 3000 genes

And Genome B has 3000 genes

And they share 1000 genes

Then J(A,B) = 1000/(2000 + 2000 + 1000) = 1/5

Genomes as sets: k-mers

Instead of using “genes” as sets, we could use “k-mers” instead.

In principle, we could break every genome into its component k-mers (say k=21), and end up with about 3 million k-mers each (for a bacterium)

That ends up being computationally expensive when you have many sets to compare:

$$\binom{N}{2} = \frac{N(N-1)}{2} \approx \frac{N^2}{2}$$

There are currently 335,736 genomes in the NCBI Genomes database.

To compare all-vs-all would require about 56 billion comparisons.

This would take a long time if the full set of k-mers was used for each comparison.

MinHash Sketches

One way to simplify this massive computational job is to approximate the Jaccard Index instead of calculating it exactly.

The MinHash algorithm does this. For DNA sequences:

1) Extract all k-mers from a sequence
2) Use a *hash function* to reproducibly convert k-mers into numbers that are randomly distributed over some range.
3) Choose the numerically lowest *s* hashes (*min hashes*) as the *bottom sketch*
4) Compare bottom sketches against each other using the Jaccard index

A *hash function* maps an input range to a fixed output range.

Example of a *bad* hash function:

1) Take a string of DNA bases:
   `ACATACTCGATCAGCTGAC`
2) Substitute 0 for A/T and 1 for C/G
   `0100010110010110101`
3) Take the first 5 digits and decimal-ize
   `1000`

This will lead to *collisions* where many different sequences will end up with the same hash. A good hash function has high compressibility, high randomization, low collision rate, and high speed.

Hashing functions

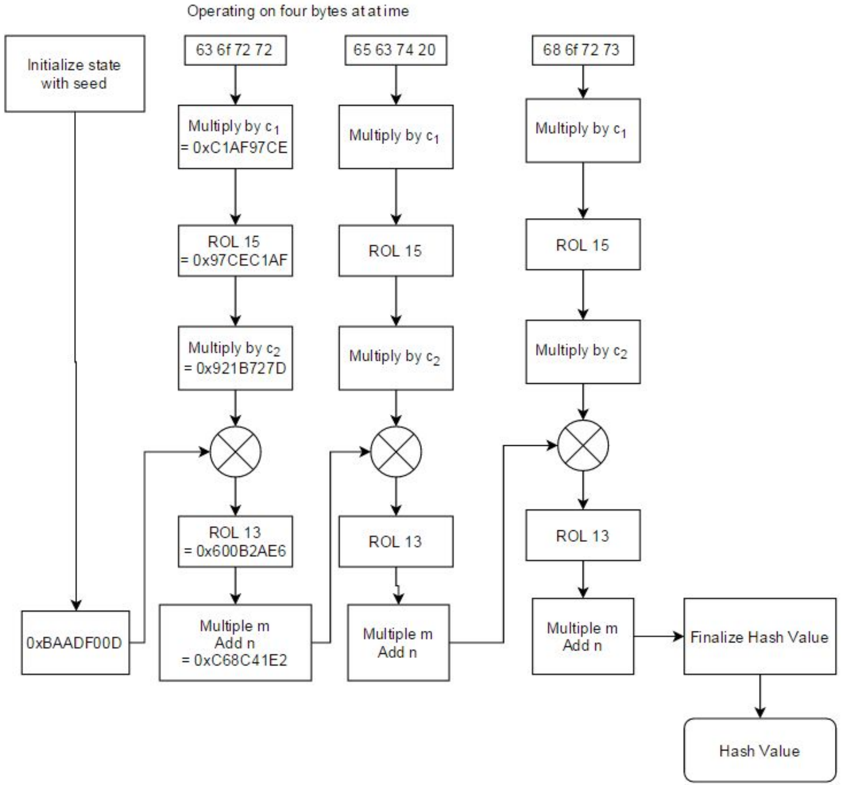
The hashing function used in most MinHash implementations is MurmurHash3. Identical sequences hashed using the same function and k-mer size will produce identical hashes.

Here are the bottom-10 sketches for two strains of *Shewanella baltica*. **Q:** What is their Jaccard Index?

575255017582
935685935937
2925290528259
1449102556507
2396553957178
2442404261705
2639034908046
4109890208483
43898691665130
47367865367011

2925290528259
7397951382043
21567977839038
2396553957178
2639034908046
43898691665130
46272541519390
46736359986916
46822418898135
47180432856748

MurmurHash3 function



Hashing functions

Q: Why not just take the lexicographically smallest *s* k-mers?

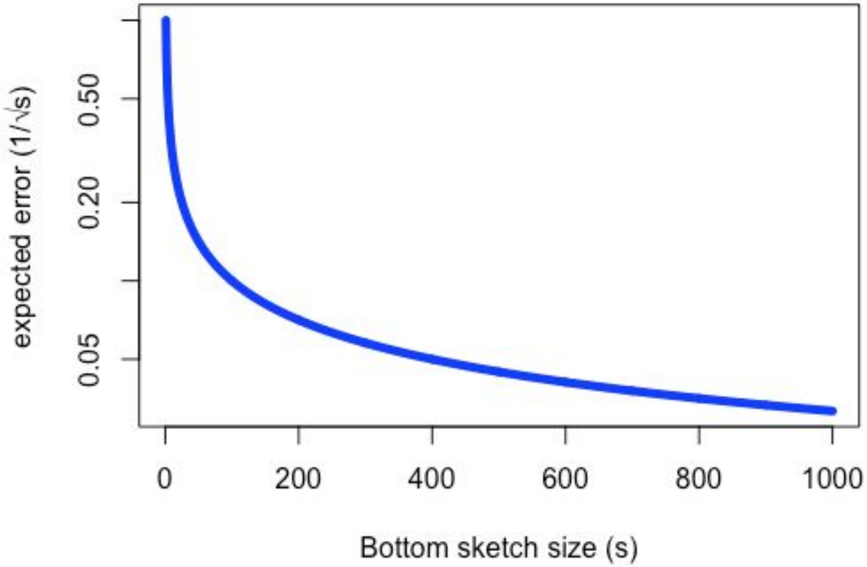
A: Because that’s not a random selection, you could have biases due to GC content, unusual genes, etc. Some implementations use a blacklist or other techniques to avoid artifacts.

Q: Why not randomly select *s* k-mers from each set and sort them lexicographically?

A: Because it’s unlikely two large sets will overlap with a small *s* unless they are highly similar. Using the bottom sketch ensures that an overlap will be found if it exists.

Q: How many signatures are needed to get a good approximation?

A: Using MinHash, the expected error rate is *e*=1/sqrt(*s*), such that *e*=0.05 at *s*=400

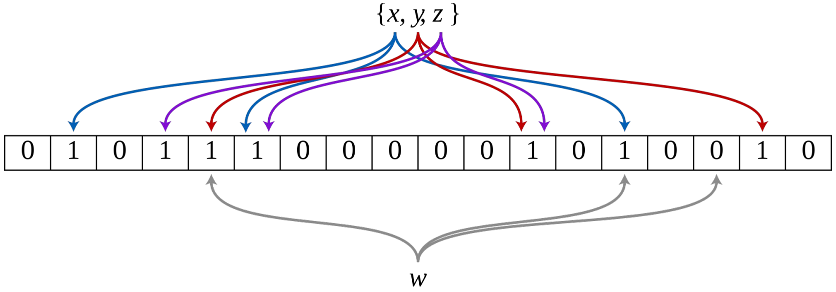


Bloom filters

A common way to speed up k-mer computations in bioinformatics is to use a (Burton Howard) Bloom filter to test whether an element is a part of a set. It trades a large memory footprint for speed on tasks like rejecting singletons or error-containing k-mers.

A Bloom filter is described by 2 parameters:

1) ***m*** = length of filter (e.g. 2^{31} bits) initialized to all 0s
2) ***n*** (<<***m***) = number of hash functions that each produce a hash in the range of the filter (e.g. 1 to 2^{31}).



Procedure:

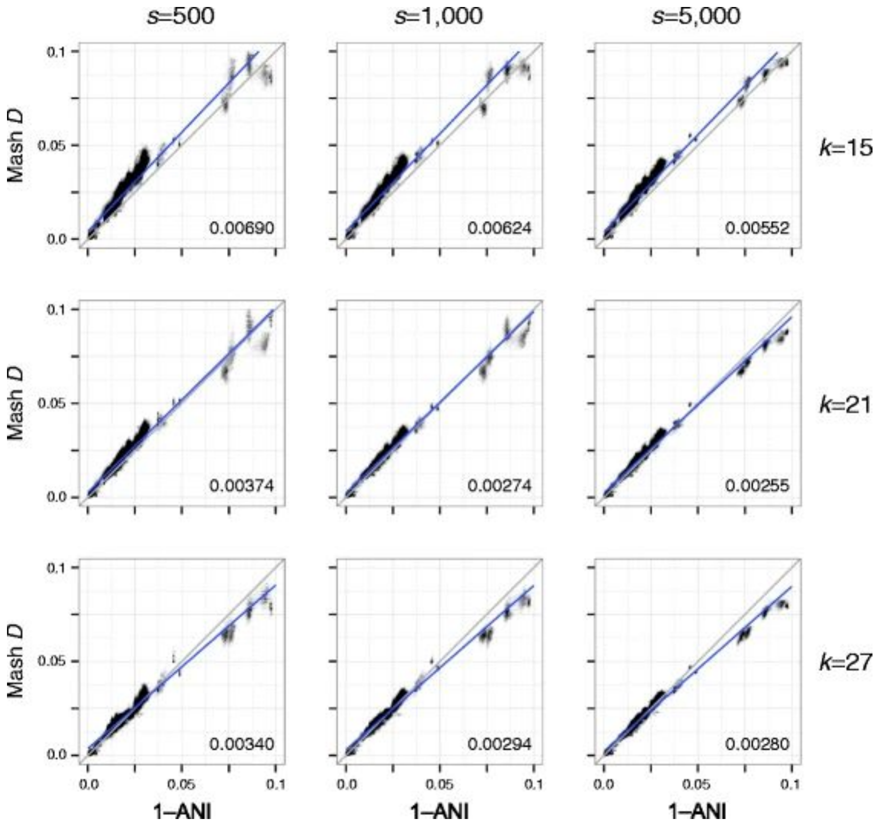
1) To insert an element: calculate all ***n*** hashes and insert them by flipping the corresponding bits to 1
2) To test whether an element is part of a set: calculate all ***n*** hashes and check the bits.
   a) If all bits are set, then “maybe”
   b) If at least 1 bit is unset, then “definitely not”

False positives are possible. False negatives are not possible. The desired false positive rate can be tuned by changing ***m*** and ***n****.*

MinHash compared to ANI

Average Nucleotide Identity (ANI) is a method to compare the similarity between core genomes. MinHash does a good job approximating ANI.

“Scatterplots illustrating the relationship between ANI and Mash distance for a collection of Escherichia genomes. Each plot column shows a different sketch size s and each plot row a different k-mer size k. Gray lines show the model relationship D = 1–ANI and numbers in the bottom right of each plot give the root-mean-square error versus this perfect model. Blue lines show linear regression models. Increasing the sketch size improves the accuracy of the Mash distance, especially for more divergent sequences. However, there is a limit on how well the Mash distance can approximate ANI, especially for more divergent genomes (e.g. ANI considers only the core genome)”



Ondov et al. (2016)

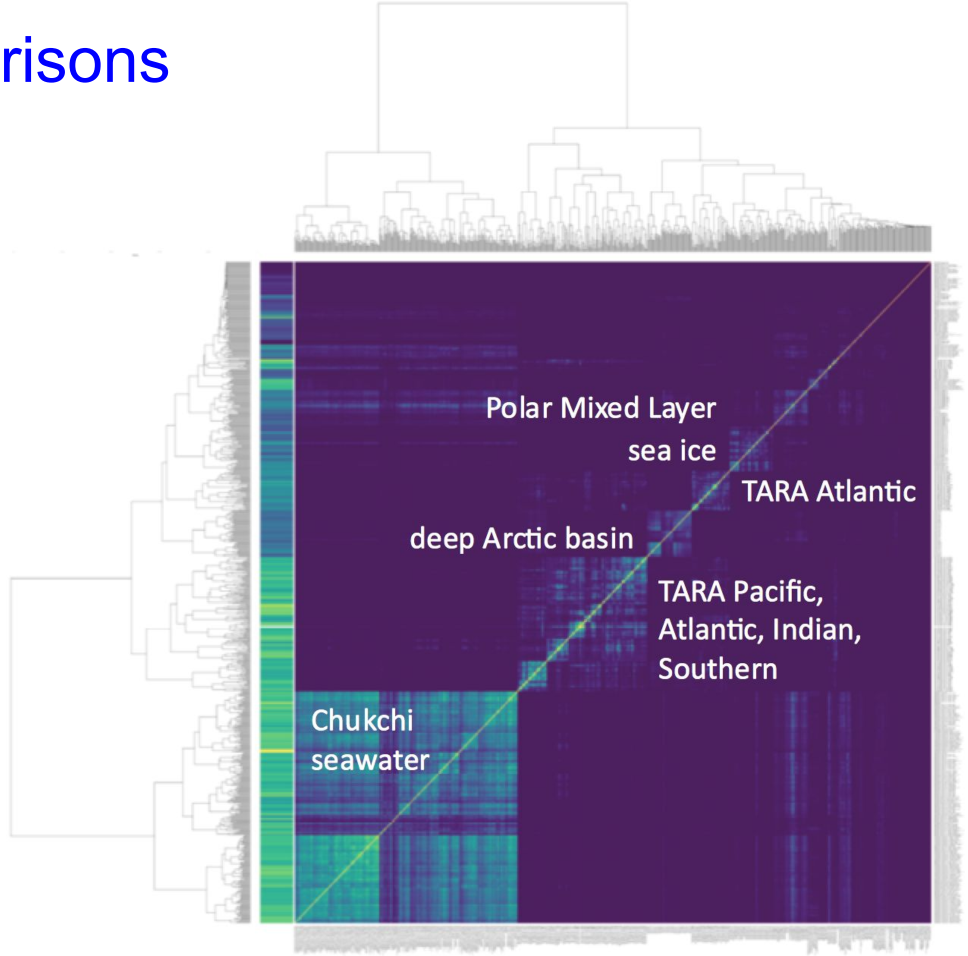
MinHash metagenome comparisons

MinHash can be used to quickly compare metagenomes for related content, including containment of a particular genome in the metagenome.

The figure on the right is a MinHash comparison of about 1000 ocean metagenomes, colored by pairwise Jaccard Index.

Sourmash has the following databases available for download to use in metagenome classification:

- 60k RefSeq microbial genomes
- 100k GenBank microbial genomes
- 87k GenBank microbial genomes LCA



MinHash implementations

- **Mash [https://github.com/marbl/mash]**
  - can use a Bloom filter to exclude singletons and other method for min k-mer count
  - Implemented online at PATRIC Similar Genome Finder
- **Sourmash [https://github.com/dib-lab/sourmash]**
  - can use a Bloom filter with counting for min k-mer count
  - can use a ‘scaled’ (proportional) *s* rather than static
  - can generate figures
  - LCA taxonomic classification
- **bbsketch [https://sourceforge.net/projects/bbmap]**
  - can use min k-mer count
  - can use optional scaled *s*
  - can use blacklist
  - LCA taxonomic classification
- **Finch [https://github.com/onecodex/finch-rs]**
  - Can use a Bloom filter for min k-mer count

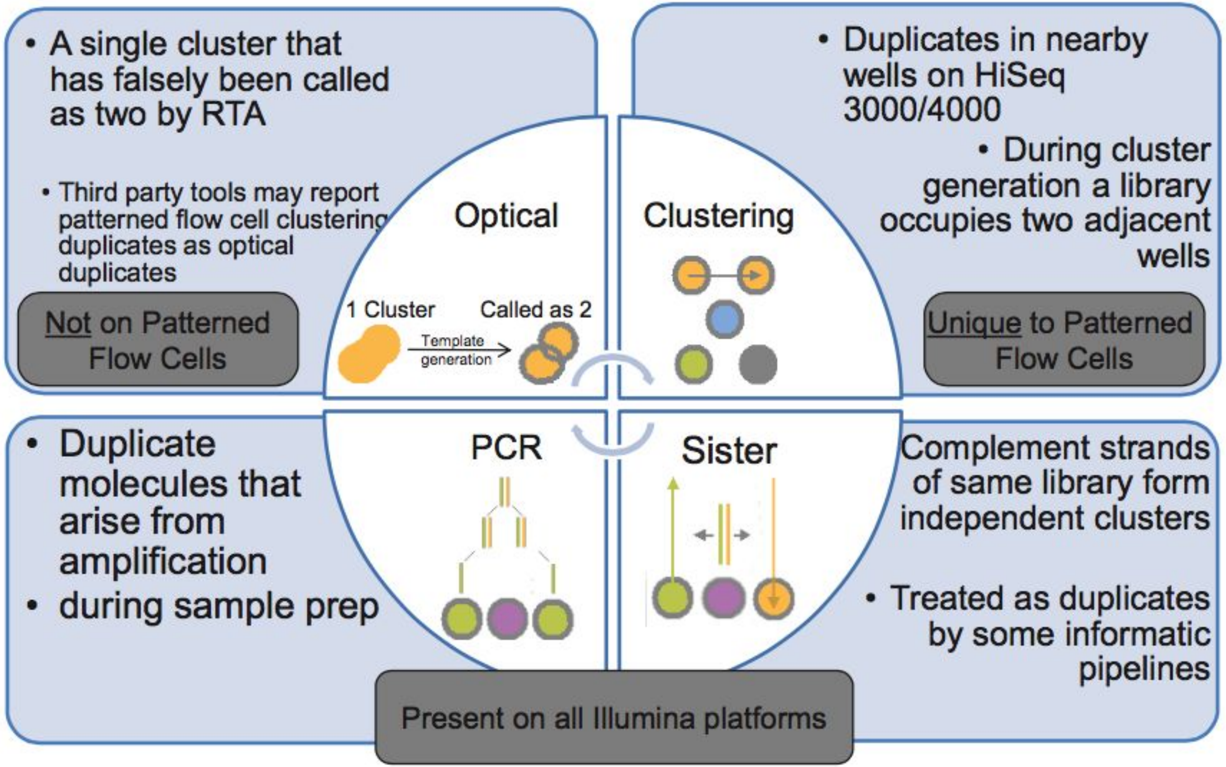
Resources

- McCormick, C. (2015, June 12). *MinHash Tutorial with Python Code*. Retrieved from http://www.mccormickml.com
- https://genomebiology.biomedcentral.com/track/pdf/10.1186/s13059-016-0997-x
- https://en.wikipedia.org/wiki/Bloom_filters_in_bioinformatics
- https://en.wikipedia.org/wiki/MinHash
- https://f1000research.com/articles/8-1006
- https://www.quora.com/What-is-the-simplest-example-of-a-hash-function
- https://www.slideshare.net/gakhov/probabilistic-data-structures-part-1-membership

Demo #3.1: sourmash, bbduk, sendsketch

- Launch binder: https://mybinder.org/v2/gh/biovcnet/metagenomics-binder-qc/master?urlpath=lab
- Click ‘Terminal’
- In the file browser on the left, navigate to **`topic-metagenomics/Lesson-3/`** and open **`Demo3.1_sourmash_bbduk_sendsketch.sh`**

Read duplicates in Illumina data



- A single cluster that has falsely been called as two by RTA
  - Third party tools may report patterned flow cell clustering duplicates as optical duplicates

Not on Patterned Flow Cells

- Duplicates in nearby wells on HiSeq 3000/4000
  - During cluster generation a library occupies two adjacent wells

Unique to Patterned Flow Cells

- Duplicate molecules that arise from amplification
- during sample prep

Complement strands of same library form independent clusters

- Treated as duplicates by some informatic pipelines

Present on all Illumina platforms

https://core-genomics.blogspot.com/2016/05/increased-read-duplication-on-patterned.html