**METAGENOMICS**

## LESSON ONE – WHAT IS METAGENOMICS

### WHAT IS METAGENOMICS?

* What it is not
  + In this house we don’t use “metagenomics” to mean “amplicon sequencing”
  + 16S “metagenomics” sequencing
* Here, we only mean “shotgun metagenomics”
  + “amplicon sequencing” or “metabarcoding” is sequencing a specific target region from many genomes (eg 16S rRNA gene)
  + “shotgun metagenomics” is (incomplete) sequencing of a mixture of genomes using an untargeted approach
* Why is shotgun metagenomics incomplete?
  + Environmental samples are information dense
  + One mL of seawater has about a million bacteria, tens of thousands of eukaryotes
    - THAT’S SO MANY BP
  + Even more genomic information in soil, and even more in stool
  + Reading truly all of that information would cost so much money... cannot be reasonably done

### WHAT KINDS OF QUESTIONS CAN METAGENOMICS BE USED TO ANSWER?

* Who is there?
  + Marker gene analysis
    - Taxonomic diversity
    - Phylogenic diversity
  + Binning
    - Taxonomic diversity
    - Phylogenic diversity
    - Novel taxa
  + Assembly
    - Genome diversity
    - Novel genomes
  + Questions
    - Is this gene present in this sample?
    - How many homologs of this gene appear in this sample?
    - Which genomes encode this gene?
    - Is this pathogen present in this environment
    - How closely related is this uncultured strain to this cultured representative
    - How many ecotypes of this bacterium appear in this environment
    - How many prokaryotic phyla are there in the world?
* What are they doing
  + Gene prediction
    - Gene diversity
    - Novel genes
  + Functional annotation
    - Protein family diversity
    - Functional diversity
  + Questions
    - What proteins do symbionts encode to mediate relationships with their hosts?
    - Which genes/pathways/genomes co-occur in this environment?
    - What antibiotic resistance genes does this community encode?
    - How many different carbon fixation pathways exist in hydrothermal vents?
    - Are there novel CRISPER-Cas systems yet to be discovered?
* Comparative metagenomics
  + Combination of who is there and what they are doing
  + Intercommunity similarity
  + Metadata correlations
  + Biomarker detection

### IS METAGENOMICS RIGHT FOR ME?

* You might tray amplicon sequencing if:
  + You need to detect rare genes of species
  + You’re working with eukaryotes
  + You have many (1000s) of samples to run
* You might try quantitative PCR if:
  + You only care about presence/absence
  + You want to quantify how many copies of a gene/species is present in a sample
* You might try isolate genomics if:
  + You can isolate your organism of interest
  + You’re working with eukaryotes
* You might try single-cell genomics if
  + Your interest is population genetics
  + Your interest is in novel taxa
  + Your interest is horizontal gene transfer and pangenomics
* You might try meta/transcriptomics if:
  + You want levels of gene expression
  + You’re working with eukaryotes

### YOU’VE GOT DATA! NOW WHAT?

* Some examples
  + metaWRAP
  + YAMP
  + ATLAS
  + SqueezeMeta

### INTERACTIVE MATERIAL

* FASTQC
  + Useage
  + fastqc file -o output.directory
* MULTIQC
  + Useage
  + multiqc directory

## LESSON TWO – TAXONOMIC CLASSIFICATION USING K-MERS

### THE POWER OF K-MERS

* What is a k-mer
  + A word of DNA that is k bases long
  + 41 k-mers: A, T, C, G
  + 42 k-mers: AA, AT, AC, AG, TA, TT, TC, TG, CA, CT, CC, CG, GA, GT, GC, GG
  + 43 k-mers: ordered combos of three
    - All codons are 3-mers but not all 3-mers are codons
* Extracting k-mers
  + Typically k-mers are extracted by running a k-length window across all of the reads and sequences
* Complementarity
  + K-mer analysis is often simplified by storing (“hashing” or “indexing”) only the lexicographically lower reverse compliment (the “canonical” string)
  + For example, you can store a particular 4mer at CGAT or ATCG, and these are equivalent
  + Standard one to choose would be whichever one would come first in the dictionary
* Palindromes
  + In molecular biology, a palindrome is a DNA sequence that is its own reverse compliment
  + Restriction enzyme cut sites are common examples of palindromes
  + Some tools (eg assemblers) require odd-length k-mers to avoid palindromes, which induce self-loops in B Brujin graphs
  + An odd-length k-mer can never be its own compliment
* Choosing a k-mer length
  + The optimal k-mer length will be determined by the goal of your analysis
  + 1-mers: GC-content analysis
    - The distribution of k-mers in the genomes is not uniform
    - Even at the level of 1-mers, bacterial genomes have different, but highly conserves proportions
    - For a given species/group, the GC content tends to be similar across the whole genome
      * This may be different than what it is for other species though
    - Could be useful for detecting HGT!
  + 3-mers: codon usage
    - At the level of 3-mers, bacterial genomes have unique, but highly conserved useage of codons
    - Extends to untranslated regions and may be primarily related to GC content
    - Some codons are preferred over other
  + 4- or 5-mers: metagenomic binning
    - 4-mers (tetranucleotide) frequency
      * At the level of 4-mers, bacterial genomes have conserved “fingerprints) which are used in metagenomic binning tools and to detect horizontal gene transfer events
      * Tend to be relatively conserved across the genome, except perhaps where HGT has occurred and in ribosomal DNA areas (because this is so highly conserved)
      * Dimensional reduction
        + Some tools: PCA, t-SNE, DBSCAN, UMAP
        + Embed data with many, many, many dimensions into a 2-D space for visualization
  + 7- to 35-mers: taxonomic classification
    - Taxonomic classification
      * As k-mer length increases, the probability of a random match between two distantly related taxa decreases exponentially
      * This can be used to classify DNA sequences based on their k-mer profiles by comparing the k-mers to a pre-computed database of taxonomically-known k-mers
      * This technique is also used to identify adapters and other contaminants in raw reads
      * Some of these tools understand that some k-mers are more useful in determining taxonomic classification than others and will analyze accordingly
  + 21- to 127-mers: genome assembly
    - Genome assembly
      * Many use Brujin graphs based on k-mer overlap
      * If the k-mer length is too short, many unrelated regions are assembled together
      * By k-mers cannot be longer then sequence lengths, thus the benefit of long-read technologies

### CHOOSING A CLASSIFIER

* Sequence format
  + Nucleotide k-mers grow at 4x
  + Amino acid k-mers grow at 20x
  + So a nucleotide 31-mer=amino acid 12-mer
* Database
  + Whole genomes
  + Marker genes
  + Ribosomal RNA genes
* Search algorithm
  + K-mer indexing
  + MinHash
  + BWT
  + BLAST-like
* Examples of tools
  + K-mers plus genomes
    - Kraken (don’t use anymore)
    - Kraken2 (nuc or aa)
    - Krakenuniq (nuc)
    - CLARK (nuc)
    - MOTHUR classify seqs (16S rRNA)
  + BWT + genomes
    - Centrifuge (nuc)
    - Kaiju (aa)
  + BWT + marker genes
    - MetaPhlAn2
    - mOTU
    - AMPHORAnet
  + Reviewed vy Breitwieser et al. (a review of methods and databases for metagenomic classification and assembly

### DEMO: ADAPTER TRIMMING

* Trimmomatic
  + Many methods to trim by, including:
    - Crop (trim to a specific length by removing base reads from the end)
    - Sliding window (once the quality slips below a threshold, cut the window)
  + Use help for full list

### DEMO: KRAKEN2

## LESSON THREE – TAXONOMIC CLASSIFICATION USING MINHASH

### MINHASH ALGORITHM

* Set similarity: jaccard index
  + We have two genomes that we’d like to caompare based on their gene content
  + A “set” is composed of a collection of genes in each genome
  + The jaccard similarity is a measure of set similarities given by the intersection of the sets divided by their union
  + Count the number of shared genes (included in both genomes), divide that by the total number of unique genes that both genomes collectively encode
* Genomes as sets: k-mers
  + Instead of using “genes” as sets, we could use k-mers
  + Break every genome into its component kmers (say k=21) and end up with about 3 million k-mers for each (in bacteria)
  + Computationally expensive when you have many sets to compare and you use the full set of k-mers
* Minhash sketches
  + One way to simplify a massive computational job
    - By approximating the jaccard index instead of calculating it exactly
    - Minhash does exactly this
  + For DNA sequences:
    - 1.) extract all k-mers from a sequence
    - 2.) use 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 and input range to a fixed output range
    - Good hash functions has high compressibility, high randomization, low collision rate (where different sequences end up with the same output from the hash function), and high speed
  + Hashing functions
    - Used most in minhash implementations is murmurhash3
    - Q and A
      * Why not take the lexicographically smallest s k-mers
        + Because that is not random selection (could have biases due to GC content, unusual genes, etc)
      * Why not randomly select s k-mers from each set and sort them lexicographically
        + Because it is unlikely that two large sets will overlap with a small s unless they are highly similar. Using the bottom sketch ensures that overlap will be found if it exists
      * How many signatures are needed to get a good approximation?
        + Using minhash, the expected error rate is e=1/sqrt(s)
        + E=0.05 at s=400
* Bloom filters
  + Common way to speed up k-mer computations in bioinformatics: burton howard bloom filter to test whether an element is a part of a set
    - Trades large memory footprint for speed on tasks like rejecting singletons or error-containing k-mers
  + A bloom filter is described by two parameters
    - m=length of filter (eg 231 bits) initialized to all 0s
    - n(<<m)=number of hash functions that each produce a hash in the range of the filter (eg a to 231)
  + procedure
    - 1.) to insert element: calculate all n hashes and insert them y flipping the corresponding bits to 1
    - 2.) test whether an element is part of set: calculated all n hashes and check the bits
      * If all bits are set, then “maybe”
      * 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
* Minhash metagenome comparisons
  + Minhash can be used to quickly compare metagenomes for related content, including containment of a particular genome in the metagenome
  + Sourmash has these databases available for use in metagenome classification:
    - 60k RefSeq microbial genomes
    - 100k GenBank microbial genomes
    - 87k GenBank microbial genomes LCA (for lowest common ancestor analyses)
* Minhash implementations
  + Mash [github.com/marbl/mash]
    - Can use a bloom filter to eclude singletons and other method for min k-mer count
    - Implemented online at PATRIC similar genome finder
  + Sourmash [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 [sourceforge.net/projects/bbmap]
    - Can use min k-mer count
    - Can use optional scaled s
    - Can use blacklist (block out certain hashes – for example those used for illumina adapters)
    - LCA taxonomic classification
  + Finch [github.com/onecodex/finch-rs]
    - Can use a bloom filter for min k-mer count

### DEMO

* Read duplicates in illumina data
  + Single cluster falsely called as two by RTA
    - Not on patterned flow cells
    - Third party tools may report patterned flow cell clustering duplicates as optical duplicates
  + Duplicates in nearby wells on HiSeq 3000/4000
    - During cluster generation, a library occupies two adjacent wells
    - Unique to patterns flow cells
  + Duplicate molecules that arise from amplification during sample prep
    - Present on all illumina platforms
  + Complement strands of same library form independent clusters
    - Treated as duplicates by some informatic pipelines
    - Present on all illumina platforms

## LESSON FOUR – READ MAPPING USING BOWTIE2

### WHAT IS READ MAPPING?

* What is read mapping?
  + Alignment of short reads to a reference genome
  + Find the area in the genome where the read matches best
  + Requires millions of string searches in a long string
* Why map reads?
  + Determine presence/absence of a genome in a sample
  + Calculate coverage as a measure of relative abundance
  + Determine variants like SNPs
  + Use coverage for binning of MAGs

### ALIGNMENT SCORING

* How does alignment work?
  + Generally:
    - 1.) find options for where a read might map using kmers
    - 2.) score the sites by number of matches, number of mismatches, and number of indels
    - 3.) pick the best hit
  + There are many scoring matrices that differ in their penalty for mismatches, gaps, and gap extensions

### DIFFERENT READ MAPPERS

* Read mapping tools
  + BWAo
  + Bowtie2
  + BBmap
  + Smalt
  + SOAP2
  + TopHat
  + And many more...
* Bowtie2:
  + Build a reference genome index
    - Typically start with genome sequence file (in fasta format)
    - Burrows-Wheeler Transform + FM index
  + Aligning sequence reads to reference genome index
    - Many parameter options
    - End to end alignment
      * Uses all bases in read
    - Local alignment
      * Can exclude bases at the ends of the read
  + Mapping quality
    - Sometimes it is challenging to identify the correct location in the reference genome (especially for reads that align to repetitive regions)
    - Indels are challenging (sometimes require a secondary realignment step; other software may be better for this task such as GATK)
    - MAPQ=mapping quality (Q=-10log10p)
      * Where p in the probability of an incorrect alignment (the read’s true origin is somewhere else in the genome)
      * High MAPQ scores are better for more confident DNA variant calling
  + Dealing with paired ends
    - Concordant pair
      * Match expectations – orientation and spacing okay
    - Discordant pair
      * Does not match expectations
      * Read 2 is an insertion element that is not in this location the reference genome
  + SAM/BAM files: the output of sequence alignment software (like bowtie2)
    - SAM: sequence alignment/map format (text-based format)
    - BAM: binary format of a SAM file, better for computation
  + SAMtools: processing SAM/BAM files post alignment
    - Sorting SAM/BAM reads by chromosome position
      * This is done prior to variant calling
    - Remove duplicate reads
      * This is often done (as these may be PCR artifacts introduced during the prep of the sequence library)

### BURROWS-WHEELER ALGORITHM

### KAIJU

* For
  + Use in a convenient online platform
  + Short read alignment
  + Converts nucleotide sequences into amino acid sequences and compares to protein library
  + Can also be used from command line!
* To use
  + Go to kaiju website
  + Name job
  + Email
  + File with sequencing reads (can be zipped)
    - Upload
    - Recommended to use the paired-end sequence as well
  + Choose reference database
  + Choose other parameters

### CENTRIFUGE

* For:
  + Short read alignment
  + Uses nucleotides
  + Use in command line (recommended to use on a server, not a personal machine, because the databases are so large)
  + Or to use in cyverse, as an app
* To use (follow steps on the website)
  + Download database
  + Build index
  + Align reads to the database

### DEMO: BOWTIE 2

## LESSON FIVE – SEQUENCE ASSEMBLY

### SEQUENCING DIVERSE SAMPLES

* Working with short-read metagenomic data
  + Do we have enough coverage to piece together a genome from a particular species?
* coverage depends on genome relative abundance
  + abundant species tend to have higher coverage
    - can depend on type of species (prokaryote vs eukaryote, etc)
    - can depend on relative genome size
    - can depend on extraction method
  + for abundant groups: draft genome assembly possible
  + for rare groups: only read-based analysis possible
* effects of evenness on coverage
  + determining the rank-abundance relationship of a community
    - before sequencing one can estimate community richness/evenness with SSU libraries, FISH, or other techniques
* gradients in diversity
  + different environments require different approaches for evaluating microbial diversity and physiology
  + high diversity habitats are more amenable to read-based analysis
  + low-diversity habitats are more amenable to assembly-based analysis
* high-complexity environments
  + targeting low-abundance population becomes very difficult without some kind of enrichment (single cell, cell sorting, etc)

### SEQUENCING STATISTICS

* what is coverage?
  + Genomes are randomly\* sheared during DNA extraction and library prep
  + During sequencing, a random subset of the fragments are sampled
  + What is the average read coverage?
    - Coverage C= LN/G
      * L in the read length
      * N is the number of reads
      * G is the haploid genome length
* POISSON DISTRIBUTION
  + Pr(X=k)=(λke-λ)/k!
    - K=times base is seen, λ=average coverage,
  + What is the probability that a base will be sequenced exactly 0 times given a coverage of 4?
    - Can use poisson to get the answer
    - Pr(X=0)=40\*exp(-4)/0!=0.0183=1.83%
    - This can help us decide what level of coverage to aim for, based on how much of the genome we are willing to leave unsequenced or sequenced fewer than a certain number of times
* Metagenome coverage: there’s always more k-mers?
  + If you keep doing more and more reads, will you always find new k-mers?
  + Will taper off eventually, but may take a TON of effort
    - Even more effort when there are errors adding k-mers
    - Even more when k-mers are short
* Metagenome coverage: always more genes and taxa?
  + Moral: sequencing only a few samples from an environment may not give you the full idea of everything that is out there... may need to do more

### ASSEMBLY PARADIGMS

* Theory
  + Read layout
  + Overlap
    - Look for regions of overlap between each read and every other read
    - Very computationally intensive
    - Matches all the overlapping fragments into the longest contiguous DNA fragment it can
    - It’s tough with heterogeneous genomes though
      * For example: diploid organism with different alleles on each copy
    - Also tough when there is undersampling (gaps in coverage)
    - Unassembled reads that do not find overlap are “singletons”
  + De Bruijin Graph
    - Break all the reads up into their k-mers
    - Instead of looking at overlaps, just looking at the kmers
    - Process
      * Index kmers in reads
      * Collect k-mers into nodes
      * Record coverage of each node
      * Rescue “mercy” k-mers that would be lost by coverage limits
      * Flag repeats and errors by anomalous coverage
      * assemble prefix and suffix nodes to create edges
      * Simplify graph by collapsing nodes into linear stretches
      * Find a path or a trail that visits each edge once
      * Traverse optimal path to reconstruct consensus sequence
  + Long read assembly
    - Long-read (multi-kbp) sequences from nanopore and PacBio suffer from high error rates, so two options good for assemblies
      * 1.) hybrid: use long reads as scaffolds to resolve repeats from the short-read assemblies
      * 2.) de novo: use high coverage of long reads to resolve errors
        + Can also use short reads for polishing completed assembly
* Overlap – layout – consensus
  + OCL software
    - Originally designed for high quality long reads from sanger sequencing
    - Used mostly for eukaryotes (?)
    - Examples
      * MIRA
      * Celera
      * MaSuRCA
      * Newbler
      * Allora
      * Arachne
      * AMOS
      * Meraculous
      * Opera
* De Bruijin Graph
  + software
    - Optimized for high-ish quality short reads from illumina sequencing
    - Used for prokaryotes and metagenomics
      * SPAdes
      * MegaHIT
      * IDBA
      * Minia
      * Ray Meta
      * SOAPdenovo
      * Velvet/Velour
      * ABySS
* Long-read
  + Long read assemblers
    - Optimized for low quality long reads from PackBio/Nanopore sequencing
    - Examples
      * Canu (Celera fork)
      * Flye
      * Miniasm/Minipolish
      * Raven
      * Redbean
      * Shasta
* Hybrid
* Meta/genome assembly
  + Parameterization + iteration = success
    - Minimum length o overlap
    - K-mer length
    - Mismatch penalty
    - Gap penalties
    - Minimum quality score
    - Contig merge stringency
    - And many more
  + The output of assembly programs include contigs, scaffolds, and assembly graphs
  + Always look at your assembly graphs!
* Genome assembly: finishedness
  + Finishedness (made-up term): size of largest connected component/total size of assembly
  + In real data, 30-40x coverage can be needed to get “perfect” assemblies due to sequencing error, biased library preparation methods, variation in DNA content due to exponential growth, or other mysterious reasons

## LESSON SIX – BINNING

### WHAT AND WHY

* What is binning
  + In an environmental sample
    - Assemble a bunch of contigs that are a mix of species
    - However, the genetic information in an environment exists in discrete cells
      * Try to separate them out like that
  + What is binning
    - Grouping sequenced and assembled DNA sequences from a metagenome into separate draft genomes
    - Currently largely restricted to bacterian and archaea
      * Streamlined genomes
      * Less sequencing is needed because they have smaller genomes
    - Possibly a “transient problem”
* Why do binning?
  + Discovery of novel organisms
    - Unlike PCR, is reference independent
      * Do not need to know even a little bit about the organism’s genome
  + Insight in trait and organism evolution
  + Predicting community interactions
  + Physiology hypothesis generation

### HOW DOES BINNING WORK

* How does binning work
  + Takes advantage of conserved features across the genome
    - Sequence composition
      * Conserved in a genome
        + At the level of 4-mers, bacterial genomes have conserved “fingerprints” which are used in metagenomic binning tools and to detect horizontal gene transfer events
        + Similar g content across an entire genome, except for the ribosome genes
    - Abundance of the DNA
      * Coverage depends on genome relative abundance
      * In practice: binning by sequence composition and abundance
    - Taxonomic affiliation (of encoded features)

### METHODS AND TOOLS

* Binning approaches: “manual”
  + Mmgenome (albertsen et al., 2013)
  + Anvi’o (Eren et al., 2015)
  + ESOM (Dick et al., 2009)
  + Composition based matric and dimensionality reduction (tSNE/PCA/UMAP)
  + Low throughput but tailored decisions per bin
  + Reproducibility can be challenging
* Binning approaches: automated
  + Many tools, often relatively easy to run
  + Use the same underlying information by weigh different
  + Work best with higher numbers of samples
  + Subject of recent large scale evaluations (eg Meyer et al, 2018)
* Possible errors in binning
* Evaluating binning approaches
* Binning approaches: metabat 2 (Kang et al., 2019)
  + Based on tetranucleotide frequency (TNF) and coverage (ABD)
  + Contigs are assigned a score S (math)
  + Builds a graph of contigs initially based on TNF
  + Contigs=nodes, similarity=edges
  + Binning by graph partitioning: modified label propagation algorithm
  + Post partitioning recruiting of: short contigs (<2500 bp) and contigs in small bins (<200k)
  + Criterion: average S similarity to contigs in bin is greater than average S similarity between contigs in bin
* Binning approaches: binsanity (Graham et al., 2019)
  + Coverage based
  + Affinity propagation: each contig is tested as “cluster center”
  + Evaluation of bin quality
  + Bin refinement with composition if needed
* Binning approaches
  + Binning aggregator
  + Scores each bin based on single copy markers
  + Picks based on score > N50 > total length
  + Rescores new bin set 🡪 repeats

### COMPLEMENTARY APPROACHES

* Complementary methods: single cell genomics
  + Sorting of individual cells and MDA amplification
  + Pros: individual genotypes, population variation, rare cells
  + Cons: poor completeness (but improving), more lab intensive, lower throughput
  + Much info available, could be its own topic
* Complementary methods: hi-c metagenomics
  + Crosslinking replicons before lysis
  + Proximity linkage: digest and religate crosslinked DNA
  + Sequence proximity linked DNA and untreated sample. Use links to bin assembly from untreated sample
  + Can resolve individual replicons with a cell
* Analogous methods: read partitioning
  + Using long k-mer presence/absence to group reads into partitions
    - Early binning tools (eg metacluster 5.0, Wang et al., 2012)
    - LSA (Cleary et al., 2015)
    - LSH – sparse coding (Kyrgyzov et al., 2020)
* Analogous methods: assembly graph query
  + Rather than “flattening the assembly graph into contigs, directly extract the relevant areas from the graph
    - (possibly) emerging approach
    - Implementation: spacegraphcats (Brown et al., 2020)
    - Allows deconvolution strain variation

### DEMO