The first midterm (March 1 in lab) covers in-class material days 1-12, labs 1-4, reading weeks 1-4 (see below for exceptions about the research papers). You may bring a 1 page (front and back), hand-written "cheat-sheet", but no other notes or resources. You will not need a calculator. I have put vocab in blue.

1. Introduction and Central Dogma
   - Three types of sequences in molecular biology: DNA, RNA, and Protein
   - How do we transition from DNA to RNA? (transcription process, start/stop codons)
   - How do we transition from RNA to Protein? (translation process, 3 bases per amino acid)
   - Do *not* need to memorize map from codons to amino acids (including start/stop)
2. Genome Assembly
   - High-level next-generation sequencing (NGS) process (obtain short reads, not entire genome)
   - What is the goal of genome assembly? What is the input; what is the output?
   - Vocab: long read, short read, base pair (bp), coverage (+ how to compute coverage)
   - Common variables: G = length of genome, n = number of reads, L = length of each read
   - What are typical values for these common variables?
   - Overlap graph assembly (often called Overlap Layout Consensus (OLC) assembly)
   - How do we detect overlaps between reads? How do we build the overlap graph? What would an ideal overlap graph look like? How can we simplify the overlap graph?
   - What is the runtime of building an overlap graph and why is it prohibitive?
   - What affect do sequencing errors and repeats have on graph-based genome assemblers?
   - De Bruijn Graph (DBG) assembly: how to build and traverse a DBG to create contigs
   - What is a k-mer and how should we choose it relative to L?
   - Additional vocab: directed multigraph, in-degree, out-degree, balanced, semi-balanced, Eulerian path/cycle, connected component
   - Two different traversal algorithms: Fleury's algorithm and the recursive algorithm
   - Time and space requirements of building and traversing a DBG
   - High-level idea (not all the details) of the modifications Velvet uses to make DBGs practical
   - Assembly evaluation: both by N50 and pairwise sequence alignment
3. Pairwise Sequence Alignment
   - What is the goal of sequence alignment? What is the input; what is the output?
   - What is the difference between local and global alignment?
   - Vocab: dynamic programming (DP), homologous, substitution, gap: insertion or deletion
   - Constructing and filling in a dynamic programming table, back-tracing to find the alignment
   - Modifications for global (Needleman-Wunsch) vs. local (Smith-Waterman) alignment

- How do we weight gaps, matches, mismatches? (BLOSUM matrix for proteins)
- Multiple ways to trace back from a given cell vs. multiple cells with max score (local only)
- Modifications to the DP algorithm to produce overlap/containment alignments
- Runtime of Needleman-Wunsch and Smith-Waterman in terms of sequence lengths

4. BWT and Read Mapping

- Why would we *not* want to use DP when aligning many short reads to a reference genome?
- What is read mapping? What is the input; what is the output?
- What is the Burrows-Wheeler Transform (BWT) of a string S? Why was it originally used?
- How can we recover the original string from the BWT? *Why* does this process work?
- How much time and space does it take to construct the BWT?
- FM-Index (BWT/L + occ + M), plus additional data structures F and A (suffix array)
- How can we use the FM-Index for exact pattern matching (i.e. the recursive formulas for start point and end point in F)? *Why* does this work?
- How do we use the suffix array A to find the pattern locations in the original string?
- What are the time and space requirements of error-free read mapping? (in terms of G, L, n)
- High-level idea of how BWA and Bowtie deal with mismatches (errors and variation)
- What does genetic variation represent? Evolutionary process of mutations on tree branches