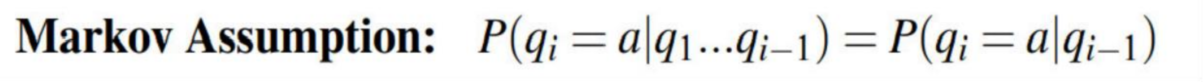
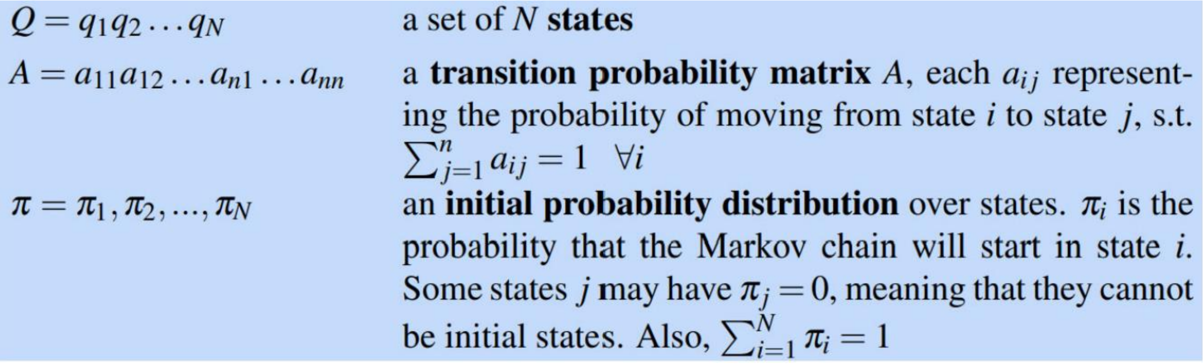
ABIN quiz Rubric

Q1: What is Markov assumption? Why is it necessary? Explain with mathematical notations. (3+1+1)

Sol:



Markov Assumption: $P(q_i = a|q_1 ... q_{i-1}) = P(q_i = a|q_{i-1})$



$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} a_{12} \ldots a_{n1} \ldots a_{nn}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{n} a_{ij} = 1 \quad \forall i$
$\pi = \pi_1, \pi_2, ..., \pi_N$	an **initial probability distribution** over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. Also, $\sum_{i=1}^{N} \pi_i = 1$

1. The probabilities of moving from a state to all others sum to one.
2. The probabilities apply to all system participants
3. The transition probabilities are constant over time.
4. The states are independent over time

Q3:

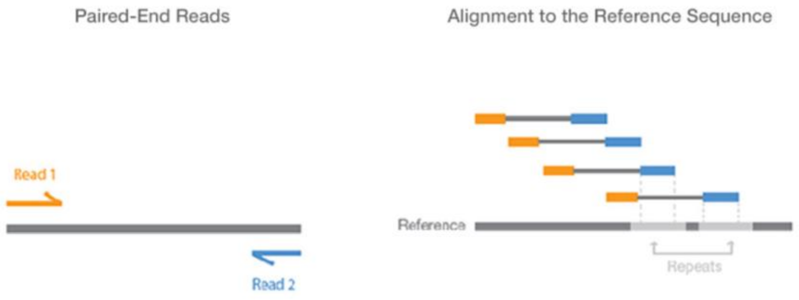
(a) What is the role of mate pair sequencing in the scaffolding step of genome assembly? Illustrate. (2+1)(2: description; 1: illustration)

Sol: Mate pair sequencing involves generating long-insert paired-end DNA libraries useful for a number of sequencing applications, including:

- *De novo* sequencing
- Genome finishing
- Structural variant detection
- Identification of complex genomic rearrangements

Combining data generated from mate pair library sequencing with that from short-insert paired-end reads provides a powerful combination of read lengths for maximal sequencing coverage across the genome.

This can be very helpful, e. g. for your De novo genome assembly. The larger inserts (mate pairs) can pair reads across greater distances. Therefore, they are able to better cover highly repetitive regions. Short-insert paired-end reads can fill in gaps missed by larger mate pairs. This combination leads to larger contigs and greater accuracy of the final consensus sequence



Paired-end sequencing enables both ends of the DNA fragment to be sequenced. Because the distance between each paired read is known, alignment algorithms can use this information to map the reads over repetitive regions more precisely. This results in much better alignment of the reads, especially across difficult-to-sequence, repetitive regions of the genome.

(b): Why does the shortest common superstring fail to reconstruct genomes from sequencing reads? (2 points)

Sol:

SCS corresponds to a path that visits every node once, minimizing total cost along path. Non-optimal superstrings can be found with a greedy algorithm At each step, the greedy algorithm “greedily” chooses longest remaining overlap, merges its source and sink.

SCS is flawed as a way of formulating the assembly problem No tractable way to find optimal SCS. Had to use Greedy-SCS. Answers might be too long.

SCS spuriously collapses repetitive sequences .Answers might be too short, by a lot!

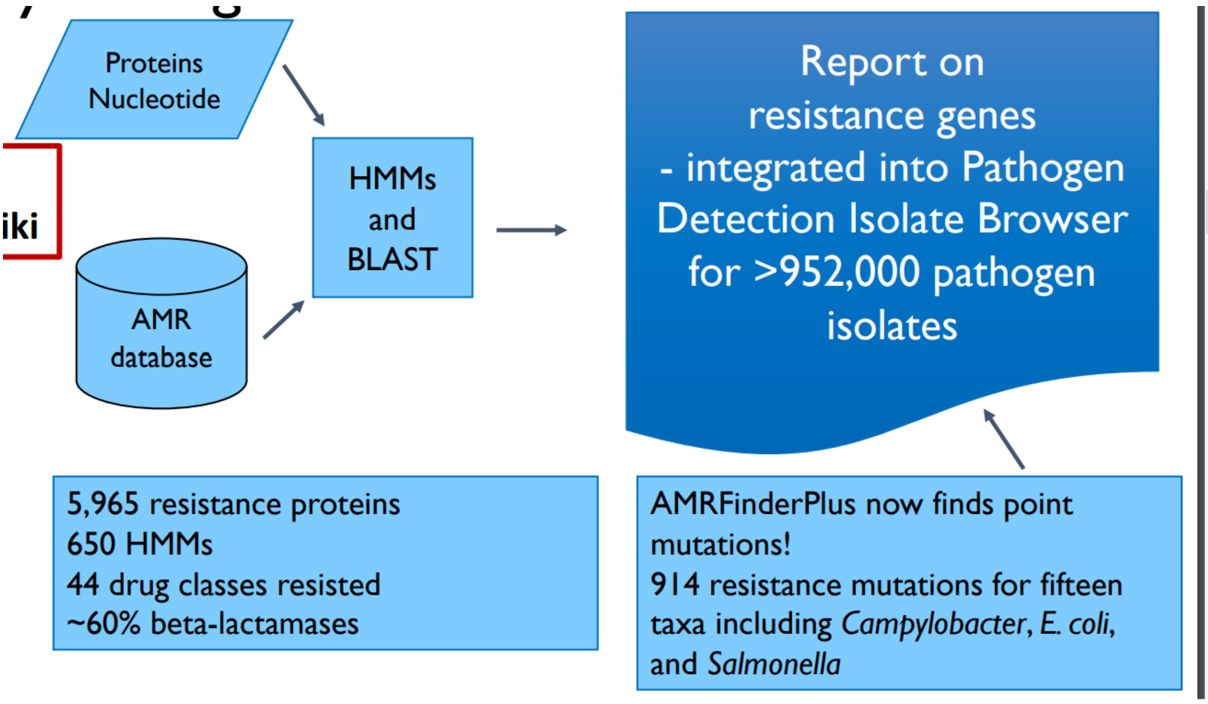
Need formulations that are (a) tractable, and (b) handle repeats as gracefully as possible

Q4: How will you use HMM to find AMR genes in bacteria? Describe steps. (5 points)

Sol:

- Alignments of known proteins are used to build HMMs that identify conserved domains of structure and function
- Typically use protein sequence for speed/computational reasons
- Based on biological structure, not arbitrary identity thresholds

1. Building an AMR database
2. The core database of profile HMM is trained using unique antibiotic resistance protein sequences.
3. After extensive curation, HMM database profiles contain profile HMMs representing major antibiotic resistance gene classes, e.g., antibiotic resistance genes against beta-lactams,
4. Use BLAST and HMMs to identify AMR genes
5. Manually curate cutoffs and threshold values
6. Report AMR genes



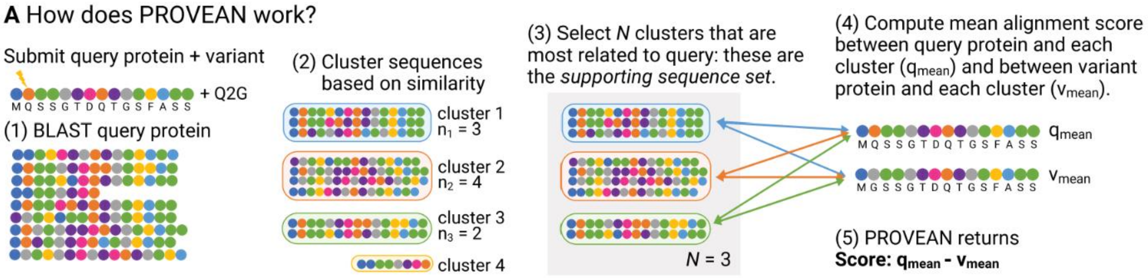
Report on resistance genes - integrated into Pathogen Detection Isolate Browser for >952,000 pathogen isolates

5,965 resistance proteins
650 HMMs
44 drug classes resisted
~60% beta-lactamases

AMRFinderPlus now finds point mutations!
914 resistance mutations for fifteen taxa including *Campylobacter*, *E. coli*, and *Salmonella*

Q5: Describe the steps of the PROVEAN algorithm for variant effect prediction. (5 points)

Sol:



A How does PROVEAN work?

Submit query protein + variant

(1) BLAST query protein

(2) Cluster sequences based on similarity

(3) Select *N* clusters that are most related to query: these are the *supporting sequence set*.

(4) Compute mean alignment score between query protein and each cluster (q_{mean}) and between variant protein and each cluster (v_{mean}).

(5) PROVEAN returns
Score: q_{mean} - v_{mean}