End of Semester Study Guide

December 10, 2025

This study guide is intended to help you to review for the final exam. This is not an exhaustive list of the topics covered in the class and there is no guarantee that these questions are representative of the questions on the exam. You should also review the notes you took in class, the notes and readings on the syllabus, and your homework assignments.

The final exam is cumulative; it covers the entire semester, but with a strong emphasis on the last third of the course. This exam is closed book. You may bring four pages (or one page, front and back) of your own notes. No electronic devices may be used during the exam. There is a clock in the class room. You will not need a calculator to answer the questions.

Pairwise sequence alignment

- Terminology: Alphabet, sequence, string, subsequence, substring.
- Dynamic programming algorithms for *local* and *global* alignment.
  - Be familiar with the basic components of these algorithms: initialization, recursion, optimal score, traceback.
  - What is the computational complexity of alignment with dynamic programing?
  - How do the basic algorithmic components differ for *local* and *global* alignment?
    - What types of scoring functions are (un)suitable for each of these?
    - Do any of the three types of alignment impose more restrictive criteria on the scoring function used? If so, what is the rationale for these criteria?
- Scoring functions
  - Similarity scoring. What are the required properties of simple similarity functions for sequence alignment? Which alignment problems can be solved with similarity scoring and which cannot? Why or why not?
  - What is edit distance? How does distance scoring differ from similarity scoring? Which alignment problems can be solved with edit distance and which cannot? Why or why not?

  - You should be able to explain how changing a scoring function will influence the nature of optimal alignments obtained with respect to that scoring function.
  - What is meant by the expected per site alignment score under a model of chance alignment? What is meant by residue background frequencies?

- Applications: Given a particular sequence analysis scenario (e.g., sequence assembly, identifying introns, etc.), you should be able to state which type of alignment is most appropriate and why.

Global multiple sequence alignment

- The canonical approach to the global multiple sequence alignment (MSA) problem is the dynamic programming approach that is used for pairwise sequence alignment. You should be familiar with the formal definition of a multiple sequence alignment, which is a direct extension of the formal definition of a pairwise alignment.

- You should understand sum-of-pairs (SP) scoring, the most common approach to scoring columns in an MSA. SP scoring is easy to work with mathematically, but overestimates the number of substitutions that gave rise to each site. Why?

- You should understand the relationship between a pair of sequences in an optimal MSA, and the optimal pairwise alignment of those sequences:

  - A multiple alignment induces pairwise alignments
  - A column in the induced pairwise alignment may contain all gaps, even though no column in the MSA contains all gaps. Why?
  - The pairwise alignment of two sequences induced by the optimal multiple alignment may not necessarily be the same as the optimal pairwise alignment of those sequences. Moreover, the induced pairwise alignment may be more biologically realistic even though it has a suboptimal score. Why?

- Global multiple alignment with dynamic programming

  - The dynamic programming algorithm for obtaining a global alignment of k sequences is a direct extension of the dynamic programming algorithm for obtaining a global alignment of 2 sequences. You should understand the initialization and recursion steps for the global multiple alignment algorithm and be able to write it down for 3 sequences.
  - Global MSA is NP-complete.
  - Given k sequences of length n, the computational complexity of the dynamic programming algorithm for global multiple alignment is $O(n^k 2^k k^2)$. You should understand how this expression is related to the steps in the multiple alignment algorithm.
  - Because of its computational complexity, the exact alignment algorithm is not recommended for $n \gtrsim 500$ or $k \gtrsim 10$. For larger problem sizes, heuristics are used.

- Many heuristics approaches are based on the idea of a "progressive alignment". The basic strategy of progressive alignment is as follows:
  - First, pairwise alignments are constructed for all pairs of sequences. This yields a $k \times k$ matrix of pairwise distances, from which a "guide tree" is constructed.
  - A multiple alignment is constructed by repeatedly merging sub-alignments. The order in which sequences/alignments are merged is determined by the guide tree. Typically, the most similar sequence pairs are merged first.
  - Sub-alignments are merged using "profile alignment". A *profile* (i.e., an alignment) of k sequences drawn from alphabet Σ is treated as though it were a single sequence of symbols from a larger alphabet, $\hat{\Sigma}$. For example, when $k = 2$ where $\hat{\Sigma} = (\Sigma \times \Sigma) \setminus \{\bar{}\}$. Treating profiles as though they were sequences makes it possible to align profiles using the pairwise dynamic programming alignment algorithm. You should understand how profile alignment works.
  - Progressive alignment follows the "once a gap, always a gap" rule. Once an alignment of a subset of the sequences is formed, it cannot be rearranged to obtain a better alignment with other sequences as they are merged into the alignment. In other words, if a bad decision is made in an early stage of the alignment process, it cannot be corrected later. As a result, progressive alignment is not guaranteed to give the optimal alignment. This policy is also the reason that progressive alignment has better time complexity than dynamic programming.
  - The computational complexity of progressive alignment is $O(k^2 n^2)$.
- The performance of MSA programs is typically evaluated using benchmarks based on curated or automated structural alignments and/or simulated sequences. Various benchmarks are designed to mimic properties of different types of data sets encountered in practice, especially those that are challenging to align:
  - Highly divergent sequences, e.g., $< 50\%$ or $< 30\%$ identity.
  - A set of closely related sequences combined with several outliers, or "orphan" sequences.
  - Related sequences that differ due to large N or C terminal extensions or large internal insertions or deletions.

Markov chains

- Definitions and terminology
  - States
  - The state probability distribution at time t
  - The initial state probability distribution.

  - The transition probability matrix. What requirements must a matrix satisfy to be a valid transition probability matrix?
  - What is the Markov property?
  - Absorbing states, reflecting states, periodic states.

- We discussed finite-state, discrete-time, time-homogeneous Markov chains. You should understand each of these terms.
- What is a periodic Markov chain? What is an irreducible Markov chain?
- n-step transitions in Markov chains: Given a transition matrix for 1 time step, you should understand how to construct a transition matrix for n time steps.
- Stationary state distributions.
  - What is the formal definition of a stationary distribution?
  - What is a limiting distribution?
  - Given the transmission matrix of Markov chain, how is the stationary distribution calculated?
  - How can you verify that a given distribution is the stationary distribution?
  - What properties may prevent a Markov chain from having a stationary distribution?
  - Under what conditions, is the stationary distribution the limiting distribution?
  - What properties are required for a Markov chain to have a unique stationary distribution?

Markov models of nucleotide substitution

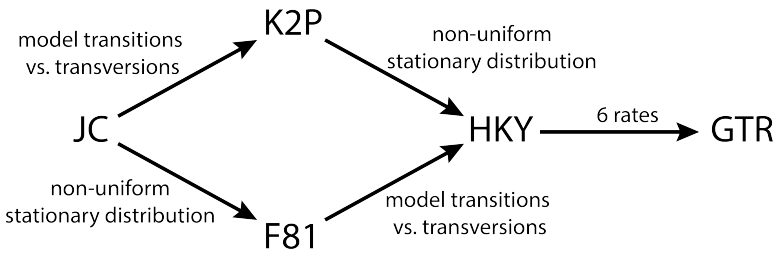
- What kinds of questions can be answered with sequence evolution models?
- In this section of the course, we focused on ungapped alignments of sequences of the same length. Why?
- What is the basic structure of a Markov model of DNA substitution?
  - What do the states of represent?
  - What is the meaning of transitions between states?
  - What are the underlying assumptions of such models?
- The Jukes Cantor (JC) model
  - What are the underlying assumptions?
  - How are transitions modeled?

  - What is the stationary distribution?
  - How is the rate parameter of the JC model related to the overall substitution rate?
  - The Jukes Cantor transition matrix gives the probability of a substitution occurring in a single time step. From this, we derived
    * the probability of observing nucleotide x at a given site and observing a different nucleotide y after elapsed time, t,
    * the probability of observing nucleotide x at a given site and observing the same nucleotide after elapsed time, t;
    * the probability of observing x aligned with y in an ungapped alignment of two sequence that have been diverging independently from a common ancestor for elapsed time t;
    * the probability of a mismatch at a given site in sequences that have been diverging independently from a common ancestor for time t;
    * the expected number of substitutions that occurred since the divergence of a pair of present-day sequences, given the number of mismatches observed in their alignment.

    You should understand each of these quantities and know how to apply them in simple scenarios. For the exam, you do not need to know how to derive these quantities.

- The Kimura 2 parameter (K2P) model
  - Kimura's model of DNA substitution distinguishes between transitions and transversions. What are transitions and transversions? Note that the word "transition" has two different meanings in this context. It is used to describe the progression from one state to another state in a Markov chain. It is also used to describe a class of nucleic acid substitutions. You should understand both of these meanings.
  - What are the underlying assumptions of K2P?
  - What are the parameters of the model?
  - What is the stationary distribution of this model?
  - The Jukes-Cantor model can be used to estimate various properties of alignments (e.g., the probability of a mismatch, corrections for multiple substitutions). These are listed above. The Kimura 2 parameter model can also be used to estimate these properties of alignments. You are not expected to have memorized the equations for the Kimura 2 parameter model, but if you were given the equations, you should be able to interpret them and understand how to apply them.

- The DNA substitution model hierarchy: We discussed a hierarchy of increasingly complex models of DNA sequence evolution. In addition to the JC and K2P models, which we discussed in detail, we considered the Felsenstein (F81) model, the Hasegawa, Kishino, Yano (HKY) model, and the General Time Reversible (GTR) model.



- For each of the five models you should understand
  - What are the underlying assumptions of the model?
  - How many parameters does the model have? What do those parameters represent?
  - What is the meaning of transitions between states in the model?
  - What is the relationship between the transition matrix and the stationary distribution of the model? How does one constrain the other?
- How are the different models related?
  - Non-uniform *transition probabilities*
    * The K2P, HKY, and GTR models all allow for different rates for different nucleic acid pairs. The K2P and HKY models distinguish between transitions and transversions. The GTR model allows for a different substitution rate for each of the six possible pairs of nucleotides.
    * The JC model assumes all substitutions proceed at the same rate.

  - Non-uniform *stationary distributions*
    * Both the JC and the K2P models have uniform stationary distributions. This distribution is an implicit consequence of the symmetric structure of the transition matrices of these models.
    * The F81, HKY, and GTR models allow for different underlying base frequencies.
  - How do models compare in terms of complexity?
  - How can you decide which model to use?
  - Given the transition matrix for a nucleic acid substitution model, can you determine which of the five models the matrix represents?

- Limitations:
  - Properties of sequence evolution that are not captured by the models we learned in class include
    * interactions between different sites in the same sequence,
    * insertions and deletions,
    * site-dependent rate variation (different rates at different sites), and
    * time-dependent rate variation (changes in rate over time).
  - What are the trade-offs associated with using a more complex models versus a less complex model?

Amino acid substitution models and matrices

- Log-odds formulation.
  - A likelihood ratio compares the probability that an observation is the outcome of a process described by hypothesis H_A, and the probability that the observation is due to chance, described by the null hypothesis, H_0. You should understand the interpretation of a likelihood ratio in the context of a pairwise alignment. What are the alternate and null hypotheses, H_A and H_0, in this context?
  - What are the advantages of using the log likelihood ratio, instead of simply the likelihood ratio?
  - How is the log likelihood ratio used to construct a scoring function for an alignment?
  - What does it mean if the likelihood ratio is less than one? Greater than one?
  - What does it mean if the log-likelihood ratio is less than zero? Greater than zero?

- Deriving amino acid substitution matrices: overview
  - Desired properties for a substitution matrix
    * Substitution matrices should be parameterized by evolutionary divergence.

    - Substitution matrices should account, directly or indirectly, for multiple amino acid replacements at the same site.
    - Substitution matrices should reflect biophysical properties. Pairs of residues with similar properties represent conservative replacements and should have higher similarity scores than pairs of residues with different properties, which represent non-conservative replacements.
  - You should understand the similarities and difference between amino acid substitution matrices and DNA substitution models. Both formalisms are representations of the process of substitution at a single site in a sequence. Compared with DNA, the amino acid alphabet is larger and the properties of the amino acids are more varied. Amino acid substitution models rely more heavily on learning parameters from data than nucleotide models.
  - You should be familiar with two families of amino acid substitution matrices: the PAM matrices and the BLOSUM matrices. Both families were derived according to the following general approach, although the details of each step differ between the two methods.
    1. Use a set of "trusted" multiple sequence alignments (ungapped) to infer model parameters.
    2. Count observed amino acid pairs in the trusted alignments, correcting for various types of sample bias.
    3. Estimate substitution frequencies from amino acid pair counts.
    4. Construct a log odds scoring matrix from substitution frequencies.

- PAM matrices use the Dayhoff Markov model of amino acid replacement.
  - The unit of divergence used is the PAM or "percent accepted mutation". What is the precise definition of this unit?
  - The PAM matrices are derived from a Markov model of amino acid replacement. What is the basic structure of this model?
  - What are the properties of the data that Dayhoff used to obtain amino acid pair counts for her model? How are those properties related to the underlying assumptions of the modeling strategy that she used?
  - How did Dayhoff derive counts from that data set?
  - How did Dayhoff account for potential sample bias in her data?
  - How did Dayhoff use the amino acid counts to derive the PAM transition matrix? How does this derivation account for differences in amino acid frequency and amino acid mutability?
  - How did Dayhoff ensure that her basic model corresponds to exactly 1 PAM of divergence?
  - How is the PAM-N model derived from the PAM-1 model?
  - How are multiple substitutions accounted for in the PAM framework?

  - How are the PAM log odds substitution matrices derived from the Dayhoff Markov model transition matrices?
  - The transition matrices are not symmetric. The substitution matrices are symmetric. What is the biological intuition associated with these observations?

- BLOSUM matrices
  - What are the properties of the data that the Henikoffs used to obtain amino acid pair counts for the BLOSUM matrices?
  - Partitioning sequences into clusters based on percent identity is a key aspect of the BLOSUM method.
    * How are the clusters used in the process of counting amino acid pairs?
    * How does the use of clusters account for sample bias?
    * How does the use of clusters lead to a family of matrices parameterized by divergence?

- Log odds substitution matrices: Both the PAM and BLOSUM substitution matrices are log-odds matrices. You should understand and be able to work with the log odds substitution matrix framework.
  - When a log odds substitution matrix is used to score an alignment, the score of the alignment also corresponds to a log likelihood ratio; what does this mean?
  - How should a positive element in a substitution matrix be interpreted in this context?
  - How should a negative element in a substitution matrix be interpreted in this context?
  - When comparing the main diagonal elements of matrices representing different amounts of evolutionary divergence, what trends would you expect to see?
  - When comparing the off-diagonal elements of matrices representing different amounts of evolutionary divergence, what trends would you expect to see?

- What are the similarities and differences between the PAM and BLOSUM matrices?
  - What are the major differences between the data used for the BLOSUM matrices and the data used for the PAM matrices?
  - What are the major differences in how sequence divergence is represented in the BLOSUM matrices compared to the PAM matrices?
  - You should be able to rank levels of sequence divergence in the two models.

- What are the similarities and differences between models of DNA sequence evolution and amino acid substitutions matrices? What is the relationship between
  - the PAM and BLOSUM models/matrices?
  - the Jukes Cantor and PAM models?
  - the Jukes Cantor, Kimura 2 Parameter, Felsenstein, and HKY models?

Blast and Searching Sequence Databases

- You should understand and be able to explain the following terminology:
  - Query
  - Database
  - Segment pair
  - Maximal segment pair (MSP)
  - High-scoring segment pair (HSP)
  - Word or w-mer
  - The word score, T
  - A "hit"
  - Distance between hits, A
  - Raw score
  - Bit score
  - The scoring (reporting) threshold, $\mathcal{S}_T$.
  - E-value and E-value ("Expect") threshold
  - Relative entropy
  - Target frequencies
- The BLAST heuristic
  - You should understand the role of each of the BLAST parameters and how the parameters influence the performance of the heuristic.
  - What is a "hit"? How were hits found in the 1990 BLAST heuristic?
  - What is meant by "false negative" and "false positive" in this context?
  - The 1990 version of BLAST did not consider alignments with gaps. What are the pros and cons of including gaps in the model? Consider both the running time and the sensitivity of the search.
  - What are the basic steps in original Blast heuristic ("Blast 90")?
  - The Blast heuristic was modified in 1997. What changes were made and how did those changes lead to increased speed and sensitivity?
  - How would increasing or decreasing w, T, A, or the reporting threshold, $\mathcal{S}_T$, influence each of the following?
    * unnecessary extensions and the speed of the heuristic
    * the number of false negatives
    * the number of false positives

  - How does Blast decide when to calculate a gapped alignment?
  - What is an extension cut-off? How and when are extension cut-offs used in the Blast heuristic?

- Karlin Altschul statistics
  - E-values
    * What is an E value? How does it differ from a p-value?
    * You should understand the equation

$$E = Km'n'e^{-\lambda \mathcal{S}} \tag{1}$$

      and be able to explain each of the variables in the equation.
    * What impact does a change in n, m or $\mathcal{S}$ have on E? You should be able to explain the intuition underlying these quantitative relationships.
  - Bit scores
    * What is a raw score? What is the normalized bit score? How are raw scores and normalized bit scores related?
    * You should understand the equation

$$E = m'n'2^{-\mathcal{S}_b} \tag{2}$$

      and be able to explain each of the variables in the equation.
    * How is the equation for E in terms of bit scores (Equation 2) related to equation for E in terms of raw scores (Equation 1)?
  - Karlin Altschul statistics provide an estimate of the number of MSPs that will be observed under a null hypothesis.
    * What is this null hypothesis?
    * What is the alternate hypothesis?
  - Karlin Altschul statistics were derived based on the assumption that the scoring matrix satisfies certain criteria. What are those criteria?
  - In Equations 1 and 2, E values are defined in terms of m' and n', the effective lengths of the query sequence and the database. What is meant by the "effective length" of the query sequence and the database? Why must these lengths be adjusted in the derivation of Karlin Altschul statistics?
  - Information theoretic aspects of BLAST
    * For a given query sequence, which factors influence which matrix will give the best discrimination between related sequences and chance similarity?
    * What is $\mathcal{H}$, the "relative entropy" of a substitution matrix ?
    * How is the relative entropy of a matrix related to the log-odds formalism?

    * What is meant by the information content of a substitution matrix? How does the information content of a matrix vary with evolutionary divergence?
    * What is the relationship between the length of the query sequence and the best choice of scoring matrix?
    * How does each of the following factors influence the difficulty of retrieving related sequences, while excluding unrelated sequences:
        · query length,
        · database size,
        · minimal alignment (MSP) length,
        · evolutionary divergence?
    * How much information is there in an alignment? You should be able to calculate the minimum information needed to retrieve meaningful matches.
    * From a theoretical perspective, what factors determine which substitution matrix is most appropriate for a particular query? From a practical perspective, how can you assess whether a particular matrix is a good choice?
    * What benefit is there to carrying out a search with the same query several times using different matrices?
    * The Blast interface allows the user to specify a restricted search set consisting of those sequences in the database that were sampled from species in a designated taxonomic group. Why might such a restriction decrease the number of sequences retrieved in a search? Why might such a restriction increase the number of sequences retrieved in a search?

Modeling Motifs and Patterns

– There are three major problems to solve in motif analysis. You are responsible for understanding this overarching framework.
    * Discovery: Given unlabeled sequences that share a conserved pattern or motif, discover the motif using unsupervised learning.
    * Modeling: Given labeled sequences that share a conserved pattern or motif, construct an abstract model that represents the frequencies of residues observed in the pattern.
    * Recognition: Given an abstract model of a motif and an unlabeled sequence, use the model to determine whether the unlabeled sequence contains the motif and/or predict the location of the motif in that sequence.
– Two major modeling approaches: Position Specific Scoring Matrices (PSSMs) and Hidden Markov models (HMMs).
    * PSSMs

        - Appropriate for ungapped, conserved motifs of fixed length, such as transcription factor binding sites.
        - Cannot model indels, variable length patterns, or positional dependences.
    * HMMs
        - Appropriate for modeling conserved motifs, as well as patterns in sequence composition, such as hydrophobic transmembrane regions.
        - Can model variable length patterns and positional dependences.

Position Specific Scoring Matrices

- Position specific scoring matrices (PSSMs)
    * A formalism for modeling ungapped multiple alignments
    * You should be familiar with each step in the calculation of a PSSM from an alignment, including the calculation of the
        1. Count matrix
        2. Frequency matrix
        3. Propensity matrix
        4. Log odds scoring matrix
    * Pseudocounts
        - What is a pseudocount?
        - What is the rationale for using pseudocounts?
        - You should understand how to construct a PSSM using pseudocounts.
    * Recognition with PSSMs: You should know how to use a PSSM to score a sliding window in an unlabeled sequence to find new instances of the motif.
    * The score of a sequence segment is analogous to a log likelihood ratio. You should understand why this is true. What are the alternate and null hypotheses represented by this likelihood ratio?
    * How are PSSMs similar to amino acid substitution matrices? How do they differ from amino acid substitution matrices?

- Limitations of PSSMs
    - You should understand the following limitations of PSSMs and be able to explain how these limitations result from the way in which PSSMs are defined.
        * PSSMs cannot model positional dependencies.
        * PSSMs are not well suited to modeling variable length patterns.
        * PSSMs cannot recognize pattern instances containing insertions or deletions.
        * Boundary detection: PSSMs are not well suited to determining the precise location of boundaries between distinct biological regions. Examples of such boundaries include the first membrane-bound amino acid in a transmembrane region, the first nucleotide in a binding site, the beginning of a gene, etc.

Hidden Markov models

- Definitions and terminology
  - A Hidden Markov model (HMM) has the following components:
    1. N states $E_1 \dots E_N$
    2. An alphabet, $\Sigma = \{\sigma_1, \sigma_2 \dots \sigma_M\}$
    3. Parameters, λ:
       (a) Initial state probability distribution vector $\pi = (\pi_i)$
       (b) Transition probability matrix a_{ij}
       (c) Emission probabilities: $e_i(\sigma)$ is the probability that state E_i emits $\sigma \in \sum$
  - An HMM is a generative model that emits a sequence $O = O_1, O_2, \dots O_T$ while passing through a sequence of states $Q = q_1, q_2, \dots q_T$. We refer to the sequence of states that emitted O as the "state path".
  - If multiple sequences are under consideration we use superscripts to distinguish them: $O^1, O^2, \dots O^k$, where $O^d = O_1^d, O_2^d, \dots O_{T_d}^d$. Similarly, multiple state paths are denoted $Q^1, Q^2, \dots$, where $Q^b = q_1^b, q_2^b, \dots q_{T_b}^b$.
  - Given a sequence $O = O_1, O_2, \dots O_T$ and a state path $Q = q_1, q_2, \dots q_T$, the joint probability of visiting the states in Q and emitting O is

$$P(O, Q|\lambda) = \pi_{q_1} \cdot e_{q_1}(O_1) \cdot a_{q_1 q_2} e_{q_2}(O_2) \cdot a_{q_1 q_2} \cdot e_{q_3}(O_3) \dots a_{q_{(T-1)} q_T} e_{q_T}(O_T).$$

  - The total probability that O was emitted by a given HMM, with parameters λ, is

$$P(O) = \sum_b P(O|Q^b, \lambda) \cdot P(Q^b|\lambda) = \sum_b P(O, Q^b|\lambda).$$

  - The sum of $P(O, Q|\lambda)$, over all sequences in Σ^* and all state paths is one:

$$\sum_d \sum_b P(O^d, Q^b|\lambda) = 1.$$

  - What is meant by the "parameters" of an HMM?
  - What is "hidden" in a Hidden Markov model?
- Hidden Markov models (HMMs) are an extension of Markov chains.
  - What properties do HMMs have in common with Markov chains?
  - What features are unique to HMMs?
  - What are the advantages of using an HMM, compared to a Markov chain?
- Motif recognition using HMMs

- HMMs can be used to answer various questions about patterns in biomolecular sequences. Given a pattern recognition problem in a new biological context, you should be able to determine which of the methods that you have learned in class can be applied to answer the question. In many cases, there may be more than one approach to answering the question. The correct approach may depend on how the HMM is designed.
- Examples of recognition questions:
  * What is the probability that a given sequence, O, was generated by the HMM? *Example:* Is the sequence a transmembrane protein?
  * What is the state path that emitted a given sequence O? Otherwise stated, the goal is to assign a state to every symbol in an unlabeled sequence, O. *Example:* Identify the cytosolic, transmembrane, and extracellular regions in the sequence. In this case, we wish to assign the labels E, M, or C to each amino acid residue in the sequence.
  * What is the probability of being in state E_i when O_t is emitted? *Example:* Is a given residue localized to the membrane?
- Calculating the total probability of a sequence, O.
  * The Forward algorithm is a dynamic program that calculates $\alpha(t,i) = P(O_1, O_2, O_3, ...O_t, q_t = E_i|\lambda)$.
    · What are the initiation, recursion, and termination steps of this algorithm?
    · What is the complexity of the Forward algorithm in terms of the the number of states and length of O?
    · Given an HMM and a sequence, O, you should know how to apply the algorithm to calculate $P(O|\lambda)$.
  * The Backward algorithm is a dynamic program that calculates $\beta(t+1,i) = P(O_{t+1}, O_{t+2}, ...O_T|q_t = E_i|\lambda)$.
    · What are the initiation, recursion, and termination steps of this algorithm?
    · What is the complexity of the Backward algorithm in terms of the the number of states and length of O?
    · Given an HMM and a sequence, O, you should know how to apply the algorithm to calculate $P(O|\lambda)$.
    · Since the Forward algorithm can be used to calculate $P(O|\lambda)$, why is the Backward algorithm needed?
  * $\alpha(t,i)$ is a joint probability. $\beta(t+1,i)$ is a conditional probability. Why are these two data structures defined differently.

* A common use of the Forward algorithm is to classify a sequence by calculating the probability that it was emitted by a particular model. Typically, we compare the likelihood of the sequence under two competing hypotheses, H_1 and H_2, using a log-likelihood ratio:

$$\log \frac{P(O|H_1)}{P(O|H_2)}.$$

  - Often, H_2 is a null hypothesis.
  - Why is it useful to consider the ratio of two likelihoods instead of merely calculating $P(O|H_1)$?
  - What is the benefit of using a *log* likelihood ratio, instead of just a likelihood ratio?

– Decoding

* Given an unlabeled sequence, the goal of decoding is to classify (i.e., label) each symbol in the sequence with its associated state. In the HMM formalism, we do this by inferring the state path that generated the sequence.
* Viterbi decoding
  - Viterbi decoding assumes that the *most probable path*, $Q^* = \text{argmax}_Q\, P(Q|O, \lambda)$ is the best estimate of the state path that emitted the sequence.
  - The Viterbi algorithm actually calculates $\text{argmax}_Q\, P(Q, O|\lambda)$, rather than $\text{argmax}_Q\, P(Q|O, \lambda)$. What is the meaning of this distinction? Why is calculating $\text{argmax}_Q\, P(Q, O|\lambda)$ an acceptable approach to finding the most probable path?
  - The Viterbi algorithm is a dynamic program that calculates $\delta(t, i)$, the probability of emitting $O_1 \ldots O_t$ via the most probable path that ends in $q_t = E_i$.
  - What are the initiation, recursion, and termination steps of this algorithm?
  - How does the traceback work?
  - What is the complexity of the Viterbi algorithm in terms of the the number of states and length of O?
  - Given an HMM and a sequence, O, you should know how to apply the algorithm to obtain Q^*.
* Posterior decoding
  - Posterior decoding assumes that the sequence of *most probable states*, $\hat{Q} = \hat{q}_1 \ldots \hat{q}_T$ is the best estimate of the state path that emitted the sequence.
  - The most probable state at time t is the state that has the highest probability of emitting O_t when all possible state paths are considered:

$$\begin{aligned} \hat{q}_t &= \operatorname*{argmax}_i P(q_t = E_i, O|\lambda) \\ &= \operatorname*{argmax}_i \alpha(t, i) \cdot \beta(t + 1, i). \end{aligned}$$

  - The most probable state, $\hat{q}$, can be estimated by using the Forward algorithm to calculate $\alpha(t, i)$ and the Backward algorithm to calculate $\beta(t + 1, i)$.

        - The sequence of most probable states may not be a valid state path; that is, it is possible that $P(O, \hat{Q}|\lambda) = 0$. How can that be?
    - Comparing Viterbi and Posterior decoding
        - Under what circumstances might posterior decoding provide a better estimate than Viterbi decoding?
        - Under what circumstances might Viterbi and posterior decoding provide the same estimate?

- Modeling and discovery with HMMs
    - Overview
        - HMM design involves two major tasks:
            1. specifying the model topology and
            2. estimating the parameters.
        - If the pattern of interest is unknown, then parameter estimation also involves motif discovery.
        - HMM design involves a trade-off between model complexity, on the one hand, and overfitting and multiple local optima, on the other. More expressive models with more parameters can capture more complex biological phenomena, but require larger training sets to obtain accurate estimates of the parameters without overfitting.
    - HMM topology
        - The HMM topology is specified by the states, $E_1, \ldots, E_N$.
        - The state connectivity is specified by defining certain transitions to have zero probability, typically to reflect boundary conditions in the biological system that the model is intended to represent. For example, in the transmembrane model, $a_{CE} \equiv 0$, because a protein cannot jump from the cytosol to the extracellular matrix without passing through the membrane.
        - One could define the model to be fully connected and allow the parameter estimation process to discover which transitions have zero probability, but this is not done in practice. What are the disadvantages of that approach?
        - Alphabet of emitted symbols (Σ): For biomolecular sequences, the alphabet will typically be $\{A, C, G, T\}$ or the twenty amino acids. However, sometimes it is convenient to use a reduced alphabet. For example, amino acid sequences can be recoded using a two letter alphabet, $\{H, L\}$, where H designates a hydrophobic amino acid and L designates a hydrophilic amino acid. A smaller alphabet reduces the number of emission probabilities to be inferred.

- Parameter estimation
  - Once the alphabet, states, and state connectivity have been chosen, the parameters of an HMM are estimated from training sequences, $O^1, O^2, ..., O^k$.
  - If the sequences are labeled, the transition and emission probabilities can be estimated from the observed transition and emission frequencies. If the sequences are unlabeled, we must first discover the conserved pattern using unsupervised learning.
  - Labeled sequences
    - If the sequences are labeled, the parameters are estimated by counting, for each state, the number of emissions and transitions observed in the data.
    - This is a form of maximum likelihood estimation (MLE).
    - You should understand the equations for estimating the initial, emission, and transition probabilities from labeled data and be able to apply them.
    - Pseudocounts can be used to account for emissions or transitions that are not observed in the training sequences. You should know how to incorporate pseudocounts in the estimation of both emission probabilities and transition probabilities.
  - Unlabeled sequences
    - If the sequences are unlabeled, then it is necessary to both discover the motif using unsupervised learning and estimate the model parameters.
    - The parameters of the model are typically learned from unlabeled data using the Baum Welch algorithm, a form of Expectation Maximization (EM).
    - Baum Welch uses an iterative, hill-climbing procedure that estimates the parameters of the model by maximizing the likelihood, $\mathcal{L}(O^1, O^2 \dots O^k|\lambda)$; that is, the probability of the training data given the parameters.
    - Baum Welch alternates between re-labeling the data from the current estimate of the parameters and re-estimating the parameters from the current labeling of the data. The labeling step uses the Forward and Backward algorithms in a modified version of Posterior decoding.
    - Baum Welch is guaranteed to converge to a local, but not a global, optimum. Executing the algorithm several times with different starting configurations can improve the chances of finding a global optimum.
    - Baum Welch estimates the parameters of the model, but does not output an explicit representation of the motif. To obtain an explicit representation of the motif, Viterbi or posterior decoding must be used to label the training sequences, once the parameters have been determined using Baum Welch.
    - The course notes give a detailed presentation of the Baum Welch algorithm (Algorithm 3 and Equations 5.8 - 5.13). We did not cover this in class and you are not responsible for the technical details on the exam. You *are* responsible for understanding the context in which Baum Welch can or should be applied; which algorithms it uses; what data it takes as input; what information it provides as output; and its limitations.

- Profile HMMs and global multiple sequence alignment
  - A Profile HMM is a specific HMM topology for modeling conserved sequence motifs, including DNA motifs representing protein binding sites and amino acid sequence motifs representing protein domains. Unlike PSSM's, a profile HMM allows for indels. (Note that although positional dependencies can be modeled using HMM's, the canonical Profile HMM topology does not capture positional dependencies between non-adjacent states.)
  - A Profile HMM of length L has $L+2$ Match states (where M_0 and M_{L+1} are silent Start and End states), L Deletion states, and $L+1$ Insertion states. What is the rationale for including $L+1$ Insertion states when there are only L non-silent Match states?
  - What is the advantage of using a Profile HMM, rather than a "custom design"?
  - You should be familiar with the Profile HMM topology and know how to apply it and interpret it. This includes how to determine the number of states and the parameters of a Profile HMM, given labeled data (i.e., a multiple alignment), and how to use a Profile HMM to find a global alignment of unaligned (i.e., unlabeled) sequences.
  - Labeled sequences:
    * Given labeled sequences, the average length of the pattern can be used as an initial estimate of the length of the model.
    * For a Profile HMM, labeled data is typically in the form of a multiple sequence alignment (MSA). The labels are implicitly specified by the columns in the alignment. A label is assigned to each column of the alignment based on the number of indels in the column.
    * You should understand the procedure for labeling columns. Each column in the MSA is labeled with an M state or an I state. What determines whether a column is labeled M or I? Columns in the MSA are never labeled with a D state. What is the rationale for this?
    * The parameters are estimated from the resulting labeled sequences by counting the symbols and transitions associated with each state.
  - Unlabeled sequences:
    * Given unlabeled sequences, use biological knowledge to obtain an initial estimate of L. Once L is chosen, the topology of the model is completely determined. It is only necessary to estimate the parameters.
    * If your initial estimate of L turns out to be a bad fit for the pattern under consideration, you can adjust the length using "model surgery". How should you assess whether the initial length estimate is appropriate for the pattern under consideration? What is model surgery and how would you apply it in a specific situation?