Study guide

December 6, 2019

This study guide is intended to help you to review for exams. 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.

Pairwise sequence alignment

- Terminology: Alphabet, sequence, string, subsequence, substring.
- Dynamic programming algorithms for *local, global* and *semiglobal* 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, global* and *semiglobal* 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
  - Edit distance. What are the required properties of distance functions for sequence alignment?
  - Similarity scoring. What are the required properties of simple similarity functions for sequence alignment?
  - 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.
- 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.

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.
- 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?
  - How can you calculate the stationary distribution of a Markov chain?
  - 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 properties is the stationary distribution the limiting distribution?
- What is the time reversibility property?
  - How do you test whether a Markov chain is time reversible?
  - Why is this property important for working with sequence evolution models?
- Simple random walk models. What are they?
  - How are they related to sequence analysis? (We discuss this last point in the BLAST lectures.)

Markov models of nucleotide substitution

- What kinds of questions can be answered with sequence evolution models?
- What are transitions and transversions?
- What is the basic structure of a DNA substitution model?
  - States,
  - Meaning of transitions?
  - Underlying assumptions?
- 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 that nucleotide x at a given site has changed to nucleotide y after elapsed time, t, as well as the probability of observing the same nucleotide at a given site after 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
  - What are the underlying assumptions?
  - How are transitions modeled?
  - What is the stationary distribution?
  - The K2P transition matrix gives the probability of a substitution occurring in a single time step. From this, we derived
    - Expressions for the probability of observing a transition, a transversion, or no change after time t has elapsed. (Given in the class notes.)
    - An expression for the expected number of substitutions of each type as a function of the number of observed transitions and transversions.

- The Felsenstein (F81) model
  - What are the underlying assumptions?
  - How are transitions modeled?
  - What is the stationary distribution?
- The Hasegawa, Kishino, Yano (HKY) model
  - What are the underlying assumptions?
  - How are transitions modeled?
  - What is the stationary distribution?
- The General Time Reversible (GTR) model.
  - What are the underlying assumptions?
  - How are transitions modeled?
  - What is the stationary distribution?
- How are the different models related?
  - Which use more complex models of nucleotide substitution? (Non-uniform *transition probabilities*)
    - The K2P, HKY, and GTR models all allow for different rates. 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 (rates are the same in both directions, i.e., A to G and G to A proceed at the same rate).
    - Both the JC and F81 models assume all substitutions proceed at the same rate.
  - Which use more complex models with 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.
  - Which models are equivalent under certain conditions? How can you derive from one model to another?
    - In contrast to the JC model, the Felsenstein model assumes all substitutions proceed at the same rate, but allows for different underlying base frequencies. How is the transition matrix in the Felsenstein model modified to achieve this?
    - The HKY model combines the innovations of the K2P and Felsenstein models to give a matrix that has different rates for transitions and transversions and allows for non-uniform base frequencies.

    - More complex models allow three or more rates. The most complex of the models within this framework is the GTR model. The GTR allows for a different substitution rate for each of the six possible pairs of nucleotides and an arbitrary stationary distribution.
    - Given an instance of the JC model and a set of non-uniform base frequencies, could you turn it into an instance of the Felsenstein model?
      More generally, given a set of non-uniform base frequencies and a transition matrix that implies uniform base frequencies, can you construct a new model that has the same rate structure as the original transition matrix, but with the specified set of non-uniform base frequencies?

- 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,
    - modelling insertions and deletions,
    - different rates at different sites (site-dependent rate variation), and
    - changes in rate over time (time-dependent rate variation).
  - Pitfalls of using more and less complex models.

Amino acid substitution models and matrices

- Log-odds formalization.
  - Know how to score an alignment given probability of a match/mismatch.
  - What does the likelihood ratio of an alignment mean? What does the log likelihood ratio of an alignment mean?
  - What does it mean if the likelihood ratio is less than/greater than 1?
  - What does it mean if the log-likelihood ratio is less than/greater than 0?
- Deriving amino acid substitution matrices: overview
  - Substitution models 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.
  - Substitution matrices should be parameterized by evolutionary divergence.
  - Given the greater number and variety of amino acids, compared with nucleotides, amino acid substitution models rely more heavily on learning parameters from data than nucleotide models.
  - 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.
- The PAM model: The Dayhoff Markov model of amino acid replacement.
  - The unit of divergence used is the PAM or "percent accepted mutation". How is the PAM defined?
  - Dayhoff's 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 Markov chain 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 this observation?

- 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 alignment score 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 divergence, what trends would you expect to see?
  - When comparing the off-diagonal elements of matrices representing different amounts of divergence, what trends would you expect to see?

- What are the similarities and differences between the PAM and BLOSUM models/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?
  - Be able to rank levels of sequence divergence in the two models.

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

- Information content of a substitution matrix
  - What is the relative entropy of a matrix?
  - How is the relative entropy of a matrix related to the log-odds formalism?
  - How does the information content of a matrix vary with evolutionary divergence?

Modelling Motifs

- Three major problems to solve
  - Discovery: Given unlabeled sequences that share a conserved pattern or motif, discover the motif using unsupervised learning.
  - Modeling: Given labeled that share a conserved pattern or motif, construct an abstract model that represents the frequencies of residues observed at each position of the motif.
  - 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 the location of the motif in that sequence.
- Two major approaches: PSSMs and 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 more loosely-defined patterns in sequence composition, such as hydrophobic transmembrane regions or CpG islands.
    - Can model variable length patterns and positional dependences.

Position Specific Scoring Matrices and the Gibbs sampler

- 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:
    1. Frequency matrix
    2. Propensity matrix
    3. Log odds scoring matrix
  - Pseudocounts
    - ∗ What are they?
    - ∗ What is the rationale for using pseudocounts?
    - ∗ Understand how to construct a PSSM using pseudocounts.
  - Recognition with PSSMs: You should know how to use a PSSM to score each position in an unlabeled sequence to find new instances of the motif.
  - The score of a window 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?

- The Gibbs sampler
  - In the context of biomolecular sequence analysis, the Gibbs sampler is a motif discovery method based on the PSSM formalism.
  - The Gibbs sampler simulates the stationary distribution of a Markov chain.
    - ∗ You should have a basic understanding of this Markov chain
    - ∗ What are the states?
    - ∗ How are states connected?
  - You should understand the basic structure of the Gibbs sampler algorithm.
  - The Gibbs sampler is guaranteed to find a globally optimal solution. What feature of the algorithm keeps it from getting trapped in local optima?
  - Even though the Gibbs sampler algorithm is guaranteed to converge to a global optimal, running the algorithm several times with different starting configurations is recommended. What is the rationale for this?
  - What is a probability density function (pdf)? What is a cumulative density function (cdf)? You should be able to calculate a cdf from a pdf.
  - You should know how to generate random numbers according to an arbitrary probability distribution, given the cdf of that distribution.

  - What are the underlying assumptions of the Gibbs sampler for biomolecular motif discovery? In what ways are they unrealistic?
  - What implementation decisions must the user make in order to apply the Gibbs sampler to a particular motif discovery problem?

- Limitations of PSSMs
  - PSSMs are designed to model fixed length conserved motifs, such as transcription factor binding sites. 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 transitions 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
  - The formal definition of an 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 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 $O^d = O^d_1, O^d_2, \dots O^d_{T_d}$. Similarly, multiple state paths are denoted $Q^1, Q^2, \dots$, where $Q^d = q^d_1, q^d_2, \dots q^d_{T_d}$.
  - 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:

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

  - What is meant by the "parameters" of an HMM?
  - What does λ usually refer to in HMM terminology?
  - What is "hidden" in a Hidden Markov model?
  - What is "decoding" and where does this term come from?
- 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 S_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 recursively calculates $\alpha(t, i) = P(O_1, O_2, O_3, ...O_t, q_t = E_i)$.
      - 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 recursively calculates $\beta(t + 1, i) = P(O_{t+1}, O_{t+2}, ...O_T|q_t = E_i)$.
      - 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?

- ∗ 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 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 this acceptable?
  - · The Viterbi algorithm is a dynamic program that recursively calculates $\delta(t, i)$, the probability of emitting $O_1 \dots O_t$ via the most probable path that ends in 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*, $\hat{Q} = \hat{q}_1 \dots \hat{q}_T$ 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:

    $$\begin{aligned} \hat{q}_t &= \underset{i}{\text{argmax}}\, P(q_t = E_i, O_t) \\ &= \underset{i}{\text{argmax}}\, \alpha(t, i) \cdot \beta(t+1, i). \end{aligned}$$

  - · 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. designing 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 and overfitting. 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, and the alphabet, Σ.
    - 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 this approach?
    - Alphabet of emitted symbols: 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. Nucleic acid sequences can be encoded in a two letter alphabet, $\{R, Y\}$, representing each base as a purine (R) or a pyrimidine (Y). Amino acids can be recoded by a six letter alphabet (e.g., one symbol for each of the so-called Dayhoff classes: AGPST, C, FWY, HRK, MILV, and NDEQ) or a two letter alphabet, $\{H, L\}$. 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 learned from unlabeled data using the Baum Welch algorithm, a form of Expectation Maximization (EM).
  - · Baum Welch uses an iterative, hill-climbing procedure to estimate the parameters of the model by maximizing $\mathcal{L}(O^1, O^2 \ldots O^k|\lambda)$, the likelihood of the data given the parameters:

$$\begin{aligned}\lambda &= \underset{\lambda_l}{\operatorname{argmax}}\, \mathcal{L}(O^1, O^2 \ldots O^k|\lambda_l) \\ &= \underset{\lambda_l}{\operatorname{argmax}} \sum_{d=1}^{k} \sum_{Q} P(O^d|\lambda_l, Q).\end{aligned}$$

  - · 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. Posterior decoding, which uses with the Forward and Backward algorithms, is used to label the data.
  - · 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.

- Profile HMMs and global multiple sequence alignment
  - A Profile HMM is a specific HMM topology for modeling conserved sequence motifs like the `WEIRD` motif. 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.)
  - A Profile HMM of length L has $L + 2$ Match states (including silent Start and End states), L Deletion states, and $L + 1$ Insertion states.
  - You should be familiar with the Profile HMM topology and know how to apply it and interpret it. In class, we discussed how to build 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.
  - The advantage of using a Profile HMM, rather than "custom design", is that once L is chosen, the topology of your model is completely determined. It is only necessary to estimate the parameters.
  - 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 a multiple alignment. The labels are implicitly specified by the columns in the alignment. A label (either a Match or an Insertion state) is assigned to each column of the alignment based on the number of indels in the column. You should understand the guidelines for deciding which label to use for each column.
    - 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 can 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?

BLAST

- Terminology and parameters
  - Query
  - Database
  - What is a segment pair?
  - What is a maximal segment pair (MSP)?
  - What is a high-scoring segment pair (HSP)?
  - Word or w-mer
  - A “hit”
  - Raw score
  - Bit score
  - Scoring threshold $\mathcal{S}_T$
  - Score T
  - Gapped and ungapped extension cut-offs X_u and X_g
  - Distance between hits A
  - Relative entropy
- 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?
  - How would increasing or decreasing w, T, A, or the reporting threshold influence each of the following?
    - the speed of the heuristic
    - the number of false negatives
    - the number of false positives
  - What problems were Gapped Blast and Two-Hit Blast designed to address?
  - The 1990 version of BLAST did not consider alignments with gaps. What are the pros and cons of including gaps in the model? Consider running time, and the sensitivity of the search.
  - What is the general process for
    - Original BLAST (BLAST90)?
    - Gapped BLAST?
    - Two-Hit BLAST?

    - combination of Gapped and Two-hit BLAST (BLAST97)?
    - PSI-BLAST?

- Karlin Altschul statistics
  - What is a raw score?
  - What is the normalized bit score?
  - How are raw scores and normalized bit scores related?
  - E-values
    - What is an E value? How does it differ from a p-value?
    - You should understand the equation

      $$E(\mathcal{S}) = \mathcal{K}\Updownarrow'\backslash'\rceil^{-\lambda\mathcal{S}} \tag{1}$$

      and be able to explain each of the variables in the equation.
    - How does E vary if one of the independent variables increases (or decreases)?
    - You should understand the equation

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

      and be able to explain each of the variables in the equation.
    - How is the equation for $E(\mathcal{S}_b)$ related to equation 1 for $E(\mathcal{S})$?
  - Karlin Altschul statistics provide an estimate of the number of MSPs that will be observed under a null hypothesis.
    - What 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?
  - What is meant by the "effective length" of the query sequence? Why must the length be adjusted in the derivation of KA statistics?
  - Information theoretic aspects of BLAST
    - What are target frequencies?
    - For a given query sequence, which factors influence which matrix will give the best discrimination between true and false positives? What is meant by true and false positives in this context?
    - What is the relative entropy of a matrix?
    - How is the relative entropy of a matrix related to the log-odds formalism?
    - How does the information content of a matrix vary with evolutionary divergence?
    - What is the relationship between the length of the query matrix and the scoring matrix used?

- ∗ Relationship between query length, database size, minimal alignment (MSP) length, and sequence divergence
- ∗ How much information is there in an alignment? You should be able to calculate the minimum information needed to retrieve meaningful matches.

Multiple sequence alignment

3 typical approaches

- Classical methods
  - Exact Dynamic Programming for MSAs
    - Appropriate for aligning sequences with gaps.
    - Mathematically optimal alignment is guaranteed.
    - NP-complete. Can only analyze a small dataset ($\approx$ 10 sequences of 500 residues)
  - Progressive Alignment
    - Appropriate for aligning sequences with gaps.
    - Not guaranteed to find the mathematically optimal alignment.
    - Much faster than Exact DP approach.
- 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 more loosely-defined patterns in sequence composition, such as hydrophobic transmembrane regions or CpG islands.
  - Can model variable length patterns and positional dependences.

Global Multiple sequence alignment using classical approaches

- You should be familiar with the formal definition of a multiple sequence alignment (MSA), 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 MSA, and a 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 is not necessarily the same as the optimal pairwise alignment of those sequences. Why?

  - Further, the induced pairwise alignment may be more biologically realistic even though it has a suboptimal score.

- The canonical approach to the global MSA problem is the dynamic programming approach that is used for pairwise sequence alignment.
  - This 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 with a maximum length of 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”.
  - You should understand the basic strategy of progressive alignment:
    1. Construct pairwise alignments for all pairs of sequences.
    2. Construct distance matrix $\mathcal{D}$, a $k \times k$ matrix of pairwise distances.
    3. Construct the “guide tree” T from $\mathcal{D}$.
    4. Construct the MSA by repeated merging in the order defined by T.
  - 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 $k = 2$, $\hat{\Sigma} = (\Sigma \times \Sigma) \setminus \{\substack{-\\-}\}$.
    * 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 $O(k^2n^2)$.

- 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.