**Study questions**

**Motif finding**

Input: string of sequences, and a kmer length

Output: kmer from each string that maximizes some score

\* Why is motif finding a challenging computational problem?

What is a good motif? Different metric yield better or worse results

Large search space: each string has n-k+1 starting positions, so there are (n-k+1)^t combinations to search

Best motif finding algorithms fail to find weak motifs in 1000 nucleotide long sequences. ChIP Seq has helped alleviate this

\* Why is the greedy motif finding algorithm we looked at unlikely to find a good solution?

The greedy algorithm for motif finding is ‘randomized motif search’, it works by guessing a profile and applying it to all strings and selecting the best strings. This makes it prone to finding local and not global minima.

This is fixed with Gibbs Sampling. This works by selecting one of the sequences at a time and withholding it, then construction the profile with the other strings/kmers. Then selecting to replace the kmer in the withheld string based on the profile of the other kmers.

\* What is a PWM and what does it represent?

PWM are probabilistic models that is used to scan a sequence to find hits for a motif, it typically implemented as a log odds score where a threshold of 50% is considered a hit.

**Traditional ML for sequence data**

\* What is the interpretation of the dot product for sequences represented as k-mer composition vectors?

The dot product is typically a distance, and so the dot product between two vectors of k-mers represents the similarity between the two vectors and their k-mers

\* How does representing fixed length sequence windows different than representing entire protein sequences?

Fixed length sequence windows is only gives you local information like motifs, but may miss more global architectures that would be preserved in global sequences

**Deep learning**

\* What is the difference in how CNNs are applied in genomics compared to computer vision?

CNN in computer are typically applied to 2D signals, whereas genomic is 1D. Additionally, images have continues values for pixels whereas the genome is one hot encoded.

\* What is the potential advantage of recurrent networks over convolutional networks?

RNN have LSTM which allow it to have memory of what it previously saw. This hopefully will allow it to capture long distant interactions, but they sadly fall short on this task which is where attention comes into play.

\* How do self-attention layers improve on convolutional and recurrent layers?  What are the drawbacks?

Self-attention is better able to capture long distant relationships in genetic sequences

\* Why are we interested in predicting DNA accessibility?

Lots of reasons, DNA is sometimes open and sometimes closed, this has large effects on downstream biology, and is how we have so many different pheontypes of cells in humans that all have the same genetic blueprint

\* Do you expect a network trained on a given set of biological samples  
  to generalize to unseen samples, e.g. when predicting protein  
  accessibility?  Similarly, do you expect a DeepBind model trained on  
  a specific ChIP-seq dataset to generalize to other ChIP-seq samples?

I do not expect to see networks trained on unseen samples to generalize well, I do expect the ability to use transfer learning to train new models more efficiently.

**Read mapping**

• Input: A string Text, a collection Patterns

containing (shorter) strings, and an integer d.

• Output: All starting positions in Text where a

string from Patterns appears as a substring with at

most d mismatches.

\* What does read coverage in an RNA-seq experiment tell us?

RNA seq read coverage show intron and exons

\* Answer the above question for other types of sequencing assays (ChIP-seq, DNase I-seq)

ChIP-seq show where proteins are binding to the genome

DNase I seq cleaves at unprotected parts of the genome and sequences them

\* What kinds of data structures are used for read mapping?

Very basically you can uses dictionaries but this has limitation, especially when there are read errors.

Trees are much more useful and tricks like adding end characters greatly improve the performance   
 two type of trees used are suffix trie but these are memory inefficient and suffix trees which improve the memory usage

 Suffix arrays

**Sequence alignment**

Symbol Matching Problem: Match as many symbols

as possible between two strings.

• Input: Two strings.

• Output: The greatest number of matched

symbols in any “alignment” of the two strings.

\* Explain in words what element $i,j$ in the dynamic programming matrix represents (very similar for either type of sequence alignment problem).

The indexes represent the position in the two sequences that we are trying to align. The i,j entry in the DP matrix is the score of the best possible alignment achieved when i, and j entries of the sequences have been aligned

\* What is a substitution matrix, and when is it used?

A substitution matrix describes the penalty implemented during alignment when a miss match occurs. It is used when implementing the dynamic programming solution to the alignment graph

\* Explain the initialization step of the Smith Waterman and Needleman Wunsch algorithms.

You need to initialize the first row and first column of the dynamic programming matrix, using the rule set up by the algorithms, which in this case is all 0s. This is because for local alignment it implements a ‘free ride’ where any starting point is valid.

## **Motif Finding**

### ****Why is motif finding a challenging computational problem?****

Motif finding involves identifying short, recurring patterns (motifs) in DNA, RNA, or protein sequences that are presumed to have biological significance. It’s challenging because:

* Motifs are often **short and degenerate** (i.e., they can tolerate mutations).
* The **signal-to-noise ratio** is low—motifs may be weakly conserved.
* There’s **combinatorial explosion**: the number of possible motifs grows exponentially with motif length and sequence count.
* Motifs may **not appear in every sequence** or may occur with **small shifts or mutations**.

### ****Why is the greedy motif finding algorithm we looked at unlikely to find a good solution?****

Greedy algorithms make the best local decision at each step, but:

* They can **get stuck in local optima** because they don't explore multiple alternatives.
* In motif finding, early choices **heavily influence** subsequent ones, so a bad early guess may lead to a poor overall result.
* They don’t backtrack or revise choices based on global context.

### ****What is a PWM and what does it represent?****

A **Position Weight Matrix (PWM)** represents a motif by describing the **frequency or probability** of each nucleotide (or amino acid) at each position in the motif. It is used to:

* Score candidate sequences for how well they match the motif.
* Visualize sequence conservation at each position.  
  PWMs assume positional independence and are constructed from aligned instances of the motif.

## **Traditional ML for Sequence Data**

### ****What is the interpretation of the dot product for sequences represented as k-mer composition vectors?****

When sequences are encoded as **k-mer composition vectors** (i.e., counts or frequencies of all k-mers), the **dot product** between two such vectors reflects:

* The **similarity** of their k-mer content.
* It’s proportional to the number of shared k-mers, weighted by their frequencies.
* In a linear model, the dot product corresponds to the **raw score before activation**, representing how much the input aligns with learned weights (like features).

### ****How does representing fixed-length sequence windows differ from representing entire protein sequences?****

* **Fixed-length windows** are used to provide consistent input sizes to machine learning models (especially neural networks). They allow **local context** learning but may ignore broader structure.
* **Entire protein sequences** vary in length and can provide **global information**, but require more complex models (e.g., RNNs, transformers) that can handle variable-length input.
* Windowed representations may **miss long-range dependencies**, while full sequences may be harder to model effectively.

## **Deep Learning**

### ****What is the difference in how CNNs are applied in genomics compared to computer vision?****

In genomics:

* CNNs operate on **1D sequences** (e.g., DNA) rather than 2D images.
* Filters (kernels) learn **motif-like patterns** instead of shapes.
* Input is often one-hot encoded (e.g., A, C, G, T → [1, 0, 0, 0], etc.)
* Tasks include binding site prediction, accessibility, and variant effect prediction.

In computer vision:

* CNNs process 2D pixel arrays.
* Filters detect edges, textures, and objects.
* Networks go deeper and use pooling to reduce spatial dimensions.

### ****What is the potential advantage of recurrent networks over convolutional networks?****

* **Recurrent Neural Networks (RNNs)** and especially **LSTMs/GRUs** can model **sequential dependencies** and **long-range interactions**, which CNNs struggle with due to local receptive fields.
* This is crucial when a nucleotide at one end of a sequence influences behavior at the other end.
* However, RNNs are **slower to train** and may suffer from **vanishing gradients** (alleviated by LSTM/GRU).

### ****How do self-attention layers improve on convolutional and recurrent layers? What are the drawbacks?****

Self-attention (used in transformers):

* Allows **all positions to interact directly**, capturing **long-range dependencies efficiently**.
* Parallelizable → **faster training** than RNNs.
* Learns **contextual representations** of sequence elements.

Drawbacks:

* **Quadratic memory/time** complexity with sequence length (expensive for long sequences).
* Requires large datasets to train well.
* Can be harder to interpret biologically.

### ****Why are we interested in predicting DNA accessibility?****

* DNA accessibility (e.g., measured by DNase I-seq or ATAC-seq) reveals which genomic regions are **open and potentially active** (e.g., promoters, enhancers).
* It helps identify **regulatory elements** and infer **gene regulatory networks**.
* Accessibility is **cell-type-specific**, so predictions can help in understanding cellular identity and function.

### ****Do you expect a network trained on a given set of biological samples to generalize to unseen samples?****

Generally, **limited generalization** is expected:

* **Biological variation** (cell type, condition, individual differences) can cause model performance to drop.
* Models like DeepBind may overfit to specific ChIP-seq data and **fail to generalize** to other datasets with different TFs, conditions, or experimental noise.
* Better generalization requires **diverse training data**, **regularization**, and sometimes **transfer learning**.

## **Read Mapping**

### ****What does read coverage in an RNA-seq experiment tell us?****

* **Read coverage** represents how many sequencing reads map to each region of the transcriptome.
* It reflects **expression levels**: more reads → higher expression.
* Coverage over exons vs. introns can also inform **splicing patterns**.

### ****Answer the above question for other types of sequencing assays (ChIP-seq, DNase I-seq)****

* **ChIP-seq**: Coverage indicates **binding intensity** of a protein (e.g., transcription factor) to DNA. Peaks in coverage = likely binding sites.
* **DNase I-seq**: Coverage indicates **chromatin accessibility**. High coverage = open chromatin; low = closed.

### ****What kinds of data structures are used for read mapping?****

Efficient read mappers rely on:

* **Suffix trees** and **suffix arrays**: Store all suffixes of the reference genome.
* **FM-index/Burrows-Wheeler Transform (BWT)**: Enables fast substring queries with low memory.
* **Hash tables** (e.g., k-mer hashing): Used in seed-and-extend mappers like BLAST.

## **Sequence Alignment**

### ****Explain in words what element**** i,ji,ji,j ****in the dynamic programming matrix represents.****

In sequence alignment (e.g., Smith-Waterman or Needleman-Wunsch), element i,ji,ji,j in the DP matrix represents the **optimal alignment score**:

* Up to position iii in the first sequence and position jjj in the second.
* It considers all possible ways of aligning the sequences up to those positions.

### ****What is a substitution matrix, and when is it used?****

* A substitution matrix (e.g., BLOSUM, PAM) provides scores for aligning each possible pair of amino acids (or nucleotides).
* It’s used to:
  + Reflect biological similarity (e.g., conservative substitutions are scored higher).
  + Guide the scoring in **protein sequence alignment** (where identities and similarities matter).

### ****Explain the initialization step of the Smith-Waterman and Needleman-Wunsch algorithms.****

* **Needleman-Wunsch (global alignment)**:
  + Initialize the first row and column with **gap penalties**: e.g., row[0][j] = -j × gap\_penalty.
  + This reflects aligning a prefix of one sequence to gaps in the other.
* **Smith-Waterman (local alignment)**:
  + Initialize the first row and column to **zero**.
  + This allows alignments to start anywhere and ensures scores don’t go below zero.