Review Session I
CS 466

Wei Qian
March 12th 2018
Topics we have covered:

- Molecular Biology
- Probability and Statistics
- Sequence and Alignment
- Pattern Matching
- BLAST
- Assembly
Definition: 4.2 **Probability distribution** of a discrete random variable

The probability distribution of a discrete random variable is the set of numbers $P(\{X = x\})$ for each value $x$ that $X$ can take. The distribution takes the value 0 at all other numbers. Notice that the distribution is non-negative. **Notation warning:** probability notation can be quirky. You may encounter $p(x)$ with the meaning “some probability distribution” or $p(x)$ meaning “the value of the probability distribution $P(\{X = x\})$ at the point $x$” or $p(x)$ with the meaning “the probability distribution $P(\{X = x\})$”. Context may help disambiguate these uses.
Definition: 4.4 **Joint probability distribution** of two discrete random variables

Assume we have two random variables \( X \) and \( Y \). The probability that \( X \) takes the value \( x \) and \( Y \) takes the value \( y \) could be written as \( P(\{X = x\} \cap \{Y = y\}) \). It is more usual to write it as

\[
P(x, y).
\]

This is referred to as the **joint probability distribution** of the two random variables (or, quite commonly, the **joint**). You can think of this as a table of probabilities, one for each possible pair of \( x \) and \( y \) values.
Definition: 4.6  The **marginal** probability of a random variable

Write $P(x, y)$ for the joint probability distribution of two random variables $X$ and $Y$. Then

$$P(x) = \sum_y P(x, y) = \sum_y P(\{X = x\} \cap \{Y = y\}) = P(\{X = x\})$$

is referred to as the **marginal** probability distribution of $X$. 
Definition: 4.8 

**Expected value**

Given a discrete random variable $X$ which takes values in the set $\mathcal{D}$ and which has probability distribution $P$, we define the expected value

$$
\mathbb{E}[X] = \sum_{x \in \mathcal{D}} x P(X = x).
$$

This is sometimes written $\mathbb{E}_P[X]$, to clarify which distribution one has in mind.
Useful Facts: 4.2  *Expectations are linear*

Write $f$, $g$ for functions of random variables.

- $\mathbb{E}[0] = 0$
- for any constant $k$, $\mathbb{E}[kf] = k\mathbb{E}[f]$
- $\mathbb{E}[f + g] = \mathbb{E}[f] + \mathbb{E}[g]$.  


Definition: 4.13 \textbf{Variance}

The variance of a random variable $X$ is

$$\text{var}[X] = E[(X - E[X])^2]$$

Useful Facts: 4.4 \textit{A useful expression for variance}

$$\text{var}[X] = \mathbb{E}[(X - \mathbb{E}[X])^2]$$
$$= \mathbb{E}[(X^2 - 2X\mathbb{E}[X] + \mathbb{E}[X]^2)]$$
$$= \mathbb{E}[X^2] - 2\mathbb{E}[X]\mathbb{E}[X] + \mathbb{E}[X]^2$$
$$= \mathbb{E}[X^2] - (\mathbb{E}[X])^2$$
Useful Facts: 4.3  Properties of variance

1. For any constant $k$, $\text{var}[k] = 0$
2. $\text{var}[X] \geq 0$
3. $\text{var}[kX] = k^2 \text{var}[X]$
4. If $X$ and $Y$ are independent, then $\text{var}[X + Y] = \text{var}[X] + \text{var}[Y]$
5. $\text{var}[X] = \text{cov}(X, X)$.

1, 2, and 5 are obvious. You will prove 3 and 4 in the exercises.
Useful Facts: 4.6 **Independent** random variables have zero covariance

1. if $X$ and $Y$ are independent, then $E[XY] = E[X]E[Y]$.
2. if $X$ and $Y$ are independent, then $\text{cov}(X, Y) = 0$.

If 1 is true, then 2 is obviously true (apply the expression of useful facts 4.5). I prove 5 below.
Definition: 2.1 *Correlation coefficient*

Assume we have $N$ data items which are 2-vectors $(x_1, y_1), \ldots, (x_N, y_N)$, where $N > 1$. These could be obtained, for example, by extracting components from larger vectors. We compute the correlation coefficient by first normalizing the $x$ and $y$ coordinates to obtain $\hat{x}_i = \frac{x_i - \text{mean}\{x\}}{\text{std}(x)}$, $\hat{y}_i = \frac{y_i - \text{mean}\{y\}}{\text{std}(y)}$. The correlation coefficient is the mean value of $\hat{x}\hat{y}$, and can be computed as:

$$\text{corr} \left( \{(x, y)\} \right) = \frac{\sum_i \hat{x}_i \hat{y}_i}{N}$$

What does 0 coefficient imply?  
What is the relationship between correlation and causality?
**Global:** Require an end-to-end alignment of $x, y$

**Semi-global (glocal):** Gaps at the beginning or end of $x$ or $y$ are free — useful when one string is significantly shorter than the other or for finding overlaps between strings.

**Local:** Find the highest scoring alignment between $x'$ a substring of $x$ and $y'$ a substring of $y$ — useful for finding similar regions in strings that may not be globally similar.
Sequence and Alignment

Global Alignment

$$OPT(i, j) = \max \begin{cases} 
\text{score}(x_i, y_j) + OPT(i - 1, j - 1) \\
sgap + OPT(i - 1, j) \\
sgap + OPT(i, j - 1) 
\end{cases}$$

Local Alignment

$$OPT(i, j) = \max \begin{cases} 
\text{score}(x_i, y_j) + OPT(i - 1, j - 1) \\
sgap + OPT(i - 1, j) \\
sgap + OPT(i, j - 1) \\
0 
\end{cases}$$
Global Alignment: DDOGC vs DOG

+1 Match; -1 Mismatch; -2 Gap.

<table>
<thead>
<tr>
<th></th>
<th>*</th>
<th>D</th>
<th>D</th>
<th>O</th>
<th>G</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>0</td>
<td>-2</td>
<td>-4</td>
<td>-6</td>
<td>-8</td>
<td>-10</td>
</tr>
<tr>
<td>D</td>
<td>-2</td>
<td>1</td>
<td>-1</td>
<td>-3</td>
<td>-5</td>
<td>-7</td>
</tr>
<tr>
<td>O</td>
<td>-4</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-2</td>
<td>-4</td>
</tr>
<tr>
<td>G</td>
<td>-6</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
</tbody>
</table>
Global Alignment: DDOGC vs DOG

+1 Match; -1 Mismatch; -2 Gap.

<table>
<thead>
<tr>
<th></th>
<th>*</th>
<th>D</th>
<th>D</th>
<th>O</th>
<th>G</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>0</td>
<td>-2</td>
<td>-4</td>
<td>-6</td>
<td>-8</td>
<td>-10</td>
</tr>
<tr>
<td>D</td>
<td>-2</td>
<td>1</td>
<td>-1</td>
<td>-3</td>
<td>-5</td>
<td>-7</td>
</tr>
<tr>
<td>O</td>
<td>-4</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-2</td>
<td>-4</td>
</tr>
<tr>
<td>G</td>
<td>-6</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
</tbody>
</table>
Global Alignment: DDOGC vs DOG

+1 Match; -1 Mismatch; -2 Gap.

<table>
<thead>
<tr>
<th></th>
<th>*</th>
<th>D</th>
<th>D</th>
<th>O</th>
<th>G</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>0</td>
<td>-2</td>
<td>-4</td>
<td>-6</td>
<td>-8</td>
<td>-10</td>
</tr>
<tr>
<td>D</td>
<td>-2</td>
<td>1</td>
<td>-1</td>
<td>-3</td>
<td>-5</td>
<td>-7</td>
</tr>
<tr>
<td>O</td>
<td>-4</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-2</td>
<td>-4</td>
</tr>
<tr>
<td>G</td>
<td>-6</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
</tbody>
</table>
# Global Alignment: DDOGC vs DOG

+1 Match; -1 Mismatch; -2 Gap.

<table>
<thead>
<tr>
<th></th>
<th>*</th>
<th>D</th>
<th>D</th>
<th>O</th>
<th>G</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>0</td>
<td>-2</td>
<td>-4</td>
<td>-6</td>
<td>-8</td>
<td>-10</td>
</tr>
<tr>
<td>D</td>
<td>-2</td>
<td>1</td>
<td>-1</td>
<td>-3</td>
<td>-5</td>
<td>-7</td>
</tr>
<tr>
<td>O</td>
<td>-4</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-2</td>
<td>-4</td>
</tr>
<tr>
<td>G</td>
<td>-6</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
</tbody>
</table>
**Global Alignment: DDOGC vs DOG**

+1 Match; -1 Mismatch; -2 Gap.

<table>
<thead>
<tr>
<th></th>
<th>*</th>
<th>D</th>
<th>D</th>
<th>O</th>
<th>G</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>0</td>
<td>-2</td>
<td>-4</td>
<td>-6</td>
<td>-8</td>
<td>-10</td>
</tr>
<tr>
<td>D</td>
<td>-2</td>
<td>1</td>
<td>-1</td>
<td>-3</td>
<td>-5</td>
<td>-7</td>
</tr>
<tr>
<td>O</td>
<td>-4</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-2</td>
<td>-4</td>
</tr>
<tr>
<td>G</td>
<td>-6</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
</tbody>
</table>

**DDOGC**

**D-OG-**
Global Alignment: DDOGC vs DOG

+1 Match; -1 Mismatch; -2 Gap.

<table>
<thead>
<tr>
<th></th>
<th>*</th>
<th>D</th>
<th>D</th>
<th>O</th>
<th>G</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>0</td>
<td>-2</td>
<td>-4</td>
<td>-6</td>
<td>-8</td>
<td>-10</td>
</tr>
<tr>
<td>D</td>
<td>-2</td>
<td>1</td>
<td>-1</td>
<td>-3</td>
<td>-5</td>
<td>-7</td>
</tr>
<tr>
<td>O</td>
<td>-4</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-2</td>
<td>-4</td>
</tr>
<tr>
<td>G</td>
<td>-6</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
</tbody>
</table>
**Local Alignment: DDOGC vs DOG**

+1 Match; -1 Mismatch; -2 Gap.

<table>
<thead>
<tr>
<th></th>
<th>*</th>
<th>D</th>
<th>D</th>
<th>O</th>
<th>G</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>O</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>
**Local Alignment: DDOGC vs DOG**

+1 Match; -1 Mismatch; -2 Gap.

<table>
<thead>
<tr>
<th></th>
<th>*</th>
<th>D</th>
<th>D</th>
<th>O</th>
<th>G</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>O</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
### Local Alignment: DDOGC vs DOG

+1 Match; -1 Mismatch; -2 Gap.

<table>
<thead>
<tr>
<th></th>
<th>*</th>
<th>D</th>
<th>D</th>
<th>O</th>
<th>G</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>O</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>
## Local Alignment: DDOGC vs DOG

+1 Match; -1 Mismatch; -2 Gap.

<table>
<thead>
<tr>
<th></th>
<th>*</th>
<th>D</th>
<th>D</th>
<th>O</th>
<th>G</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>O</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>
# Local Alignment: DDOGC vs DOG

+1 Match; -1 Mismatch; -2 Gap.

<table>
<thead>
<tr>
<th></th>
<th>*</th>
<th>D</th>
<th>D</th>
<th>O</th>
<th>G</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>O</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

---

**Alignment:**

```
-DOG-
```

---

**Sequence:**

```
DDOGC
```

**Match:**

```
D
```

**Mismatch:**

```
O
```

**Gap:**

```
G
```
Global vs. Semi-Global

Global

\[
\text{OPT}(i,j) = \max \begin{cases} 
\text{score}(x_i, y_j) + \text{OPT}(i-1, j-1) \\
\text{s}\text{gap} + \text{OPT}(i-1, j) \\
\text{s}\text{gap} + \text{OPT}(i, j-1) 
\end{cases}
\]

Base case: \( \text{OPT}(i,0) = i \times \text{s}\text{gap} \)

Traceback starts at \( \text{OPT}(n,m) \)

Semi-global (“fitting”)

\[
\text{OPT}(i,j) = \max \begin{cases} 
\text{score}(x_i, y_j) + \text{OPT}(i-1, j-1) \\
\text{s}\text{gap} + \text{OPT}(i-1, j) \\
\text{s}\text{gap} + \text{OPT}(i, j-1) 
\end{cases}
\]

Base case: \( \text{OPT}(i,0) = 0 \)

Traceback starts at \( \max_{0<j \leq n} \text{OPT}(j,m) \)
Complexity?
### Complexity?

<table>
<thead>
<tr>
<th></th>
<th>Time:</th>
<th>Space:</th>
<th>Backtrace</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$O(nm)$</td>
<td>$O(nm)$</td>
<td>$O(n+m)$</td>
</tr>
</tbody>
</table>
Sequence and Alignment

Scoring function and BLOSUM matrix

\[ c_{AB} = \sum_i c_{AB}^{(i)} \quad T = \sum_{A \geq B} c_{AB} \quad q_{AB} = \frac{c_{AB}}{T} \quad p_A = q_{AA} + \sum_{A \neq B} q_{AB} \]

\[ e_{AB} = \begin{cases} (p_A)(p_B) = (p_A)^2 & \text{if } A = B \\ (p_A)(p_B) + (p_B)(p_A) = 2(p_A)(p_B) & \text{otherwise} \end{cases} \]

score = log odds ratio = \[ s_{AB} = \text{Round} \left( \frac{1}{\lambda} \log_2 \left( \frac{q_{AB}}{e_{AB}} \right) \right) \]

rounding factor
Affine Gap Penalty

\[ g_{start} = \text{the cost of starting a gap} \]
\[ g_{extend} = \text{the cost of extending a gap by one more space} \]
\[ g_{score}(k) = g_{start} + (k-1) \times g_{extend} \]
Pattern Matching

**Goal:** Given a set of patterns and a text, find all occurrences of any of patterns in text

**Input:** k patterns \( p^1, \ldots, p^k \), and text \( t = t_1 \ldots t_m \)

**Output:** Positions \( 1 \leq i \leq m \) where substring of \( t \) starting at \( i \) matches \( p_j \) for \( 1 \leq j \leq k \)

**Motivation:** Searching database for known multiple patterns
Pattern Matching

Naive Approach

- K: number of patterns
- N: average length of pattern
- M: length of the query string

Running Time:

$O(KMN)$
Pattern Matching

Keyword Tree

- K: number of patterns
- N: average length of pattern
- M: length of the query string

Running Time:

$O(\text{KN} + \text{NM})$
Pattern Matching

**Aho-Corasick**

- K: number of patterns
- N: average length of pattern
- M: length of the query string

Running Time:

$$O(KN + M)$$
Aho-Corasick

One more example:

Preprocessing (Hash Index)

\[
\begin{align*}
\text{ATC} & \overset{h}{\rightarrow} \text{address1} & \overset{\text{retrieve}}{\rightarrow} & \{1,6,100,2000,5454, \ldots, \} \\
\text{AAA} & \overset{h}{\rightarrow} \text{address2} & \overset{\text{retrieve}}{\rightarrow} & \{15,21,30,785,3434, \ldots, \} \\
\text{TTC} & \overset{h}{\rightarrow} \text{address3} & \overset{\text{retrieve}}{\rightarrow} & \{5,164,220,502,943, \ldots, \}
\end{align*}
\]
1st. List possible match for the short word

The parameters $w$ & $T$
2nd. Find the possible matching sequences
3rd. Extend and score the match

\[ HSP = \text{High Scoring Segment Pair} \]
Statistic for choosing S

- Sequence DB
- Hits list
  - Score A
  - Score B
- EVD
- Random DB (smaller)

Score A: is significant
Score B: is NOT significant

p-value = 0.45  p-value = 0.001
P-value and E-value

In a database of size $N$: $P \times N = E$

- **P-value:**
  Probability that an alignment with this score occurs by chance in a database of size $N$.
  The closer the P-value is towards 0, the better the alignment

- **E-value:**
  Number of matches with this score one can expect to find by chance in a database of size $N$.
  The closer the E-value is towards 0, the better the alignment
Shortest Common Superstring

CTAGGCCCTCAATTTTT
CTCTAGGCCCTCAATTTTT
GGCTCTAGGCCCTCAATTTTT
CTCGGCTCTAGGCCCTCAATTTTT
TATCTCGACTTATGGCCCTCA
TATCTCGACTTATGGCC
TCTATATCTCGGCTCTAGG
GGCGTCTATATCTCG
GGCGTCTATATCT
GGCGTCTATATCT
Overlap Graph

Imagine a modified overlap graph where each edge has cost \(- (\text{length of overlap})\).

SCS corresponds to a path that visits every node once, minimizing total cost along path.

That's the Traveling Salesman Problem (TSP), which is **NP-hard**!
Assembly

de Bruijn Graph

AAABBBA

take all 3-mers: AAA, AAB, ABB, BBB, BBA

form L/R 2-mers: AA, AA, AA, AB, AB, BB, BB, BB, BB, BB, BA
Assembly

de Bruijn Graph
Eulerian Path

(a) 

(b) 

(c)
Eulerian Path

- Need to add an edge to connect the two semi-balanced nodes (if there are ones)
- Complexity?
  - $O(|E|)$
- In exam?
  - Observable
- Pseudo Code:

```python
# Make all nodes balanced, if not already
tour = []
# Pick arbitrary node
src = g.iterkeys().next()

def __visit(n):
    while len(g[n]) > 0:
        dst = g[n].pop()
        __visit(dst)
        tour.append(n)

__visit(src)
# Reverse order, omit repeated node
tour = tour[::-1][:-1]

# Turn tour into walk, if necessary
```
Good luck!