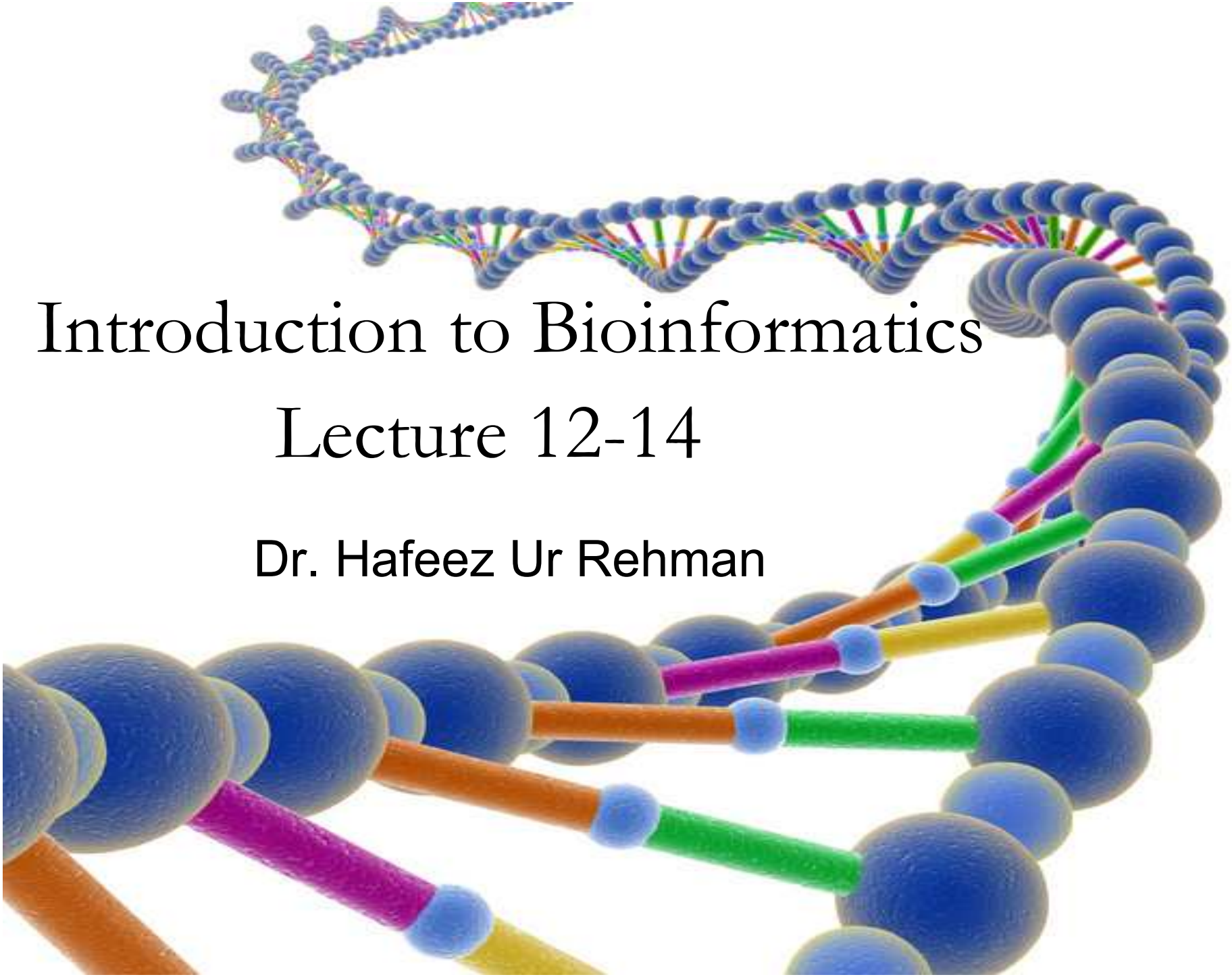


# Introduction to Bioinformatics

## Lecture 12-14

Dr. Hafeez Ur Rehman

# Heuristic alignment motivation

- $O(mn)$ too slow for large databases with high query traffic
- Heuristic methods do fast approximation to dynamic programming
  - FASTA [Pearson & Lipman, 1988]
  - BLAST [Altschul *et al*., 1990; Altschul et al., *Nucleic Acids Research* 1997]

# Heuristic alignment motivation

- Consider the task of searching the RefSeq collection of sequences against a query sequence:
  - Most recent release of DB contains 17,368,769 proteins, 2,700,925 genomic sequences
  - Entails 17,000,000*(600*600) matrix operations (assuming query sequence is of length 600 and avg. sequence length is 600)
- Need faster methods to search databases in practice

# Heuristic Algorithms

- Heuristic algorithms are based on
  - PAM and
  - BLOSAM scoring matrices.

# FASTA

Five steps are involved in FASTA:

1. Use the look-up table to identify $k$-tuple identities between I and J.
2. Score diagonals containing $k$-tuple matches, and identify the ten best diagonals in the search space.
3. Rescore these diagonals using an appropriate scoring matrix (especially critical for proteins), and identify the subregions with the highest score (initial regions).
4. Join the initial regions with the aid of appropriate joining or gap penalties for short alignments on offset diagonals.
5. Perform dynamic programming alignment within a band surrounding the resulting alignment from step 4.

# FASTA - EXAMPLE

- CONSIDER THE FOLLOWING SEQUENCE PAIR:

J = C C A T C G C C A T C G

I = G C A T C G G C

# FASTA – STEP 01

**Table 7.2.** $k$-word lists for J = CCATCGCCATCG and I = GCATCGGC, $k = 2$.

| J | | I | |
|---|---|---|---|
| AA | | AA | |
| AC | | AC | |
| AG | | AG | |
| AT | 3, 9 | AT | 3 |
| CA | 2, 8 | CA | 2 |
| CC | 1, 7 | CC | |
| CG | 5, 11 | CG | 5 |
| CT | | CT | |
| GA | | GA | |
| GC | 6 | GC | 1, 7 |
| GG | | GG | 6 |
| GT | | GT | |
| TA | | TA | |
| TC | 4, 10 | TC | 4 |
| TG | | TG | |
| TT | | TT | |

# FASTA – STEP 02

- SCORE DIAGONALS

$i = 1$, GC $\quad L_{GC}(J)=\{6\}$ $\quad l = 1 - 6 = -5$

$S_{-5} = 0 \rightarrow S_{-5} = 0 + 1 = 1$

$i = 2$, CA $\quad L_{CA}(J)=\{2,8\}$ $\quad l = 2 - 2 = 0$

$S_0 = 0 \rightarrow S_0 = 0 + 1$, and

$l = 2 - 8 = -6$

$S_{-6} = 0 \rightarrow S_{-6} = 0 + 1$

$i = 3$, AT $\quad L_{AT}(J)=\{3,9\}$ $\quad l = 3 - 3 = 0$

$S_0 = 1 \rightarrow S_0 = 1 + 1 = 2$

$l = 3 - 9 = -6$

$S_{-6} = 1 \rightarrow S_{-6} = 1 + 1 = 2$

$i = 4$, TC $\quad L_{TC}(J)=\{4,10\}$ $\quad l = 4 - 4 = 0$

$S_0 = 2 \rightarrow S_0 = 2 + 1 = 3$

$l = 4 - 10 = -6$

$S_{-6} = 2 \rightarrow S_{-6} = 2 + 1 = 3$

$i = 5$, CG $\quad L_{CG}(J)=\{5,11\}$ $\quad l = 5 - 5 = 0$

$S_0 = 3 \rightarrow S_0 = 3 + 1 = 4$

$l = 5 - 11 = -6$

$S_{-6} = 3 \rightarrow S_{-6} = 3 + 1 = 4$

# FASTA – STEP 02

- SCORE DIAGONALS

$i = 6$, GG $\quad L_{GG}(J) = \{\varnothing\}$ $\quad L_{GG}(J)$ is the empty set: no sums are increased

$i = 7$, GC $\quad L_{GC}(J) = \{6\}$ $\quad l = 7 - 6 = 1$

$S_1 = 0 \rightarrow S_1 = 0 + 1 = 1$

The final result for scores of the diagonals at various offsets is

| $\ell$ | -10 | -9 | -8 | -7 | -6 | -5 | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $S_\ell$ | 0 | 0 | 0 | 0 | 4 | 1 | 0 | 0 | 0 | 0 | 4 | 1 | 0 | 0 | 0 | 0 | 0 |

# FASTA – STEP 03

- RESCORE DIAGONALS

In step 3, we rescore the diagonals using a scoring table and allowing identities shorter than $k$-tuple length. We retain the subregions with the highest scores. The need for this rescoring is illustrated by the two examples below.

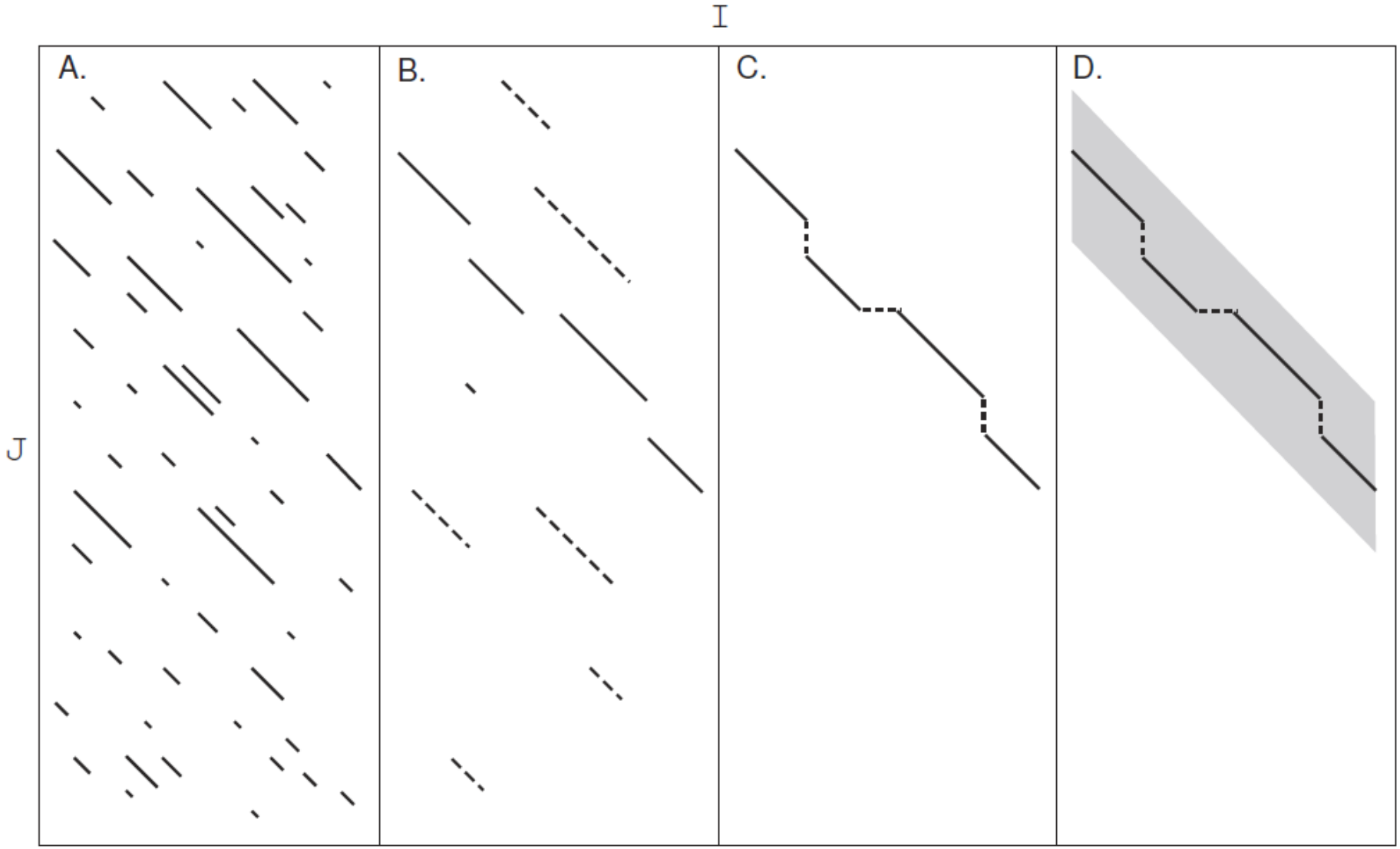
```
I:   C C A T C G C C A T C G        (Number 4-tuple matches: 0)
J:   C C A A C G C A A T C A

I':  C C A T C G C C A T C G
J':  A C A T C A A A T A A A
```

In the first case, the placement of mismatches spaced by three letters means that there are no 4-tuple matches, even though the sequences are 75% identical. The second pair shows one 4-tuple match, but the two sequences are only 33% identical. Rescoring reveals sequence similarity not detected because of the arbitrary demand for uninterrupted identities of length $k$.

# FASTA – STEP 04 & 05

- JOIN DIAGONALS AND APPLY DP



Thank you for your attention!

Questions?