CS2220: Introduction to Computational Biology

Multiple Sequence Alignment

Wong Limsoon

Outline

What & why of multiple sequence alignment

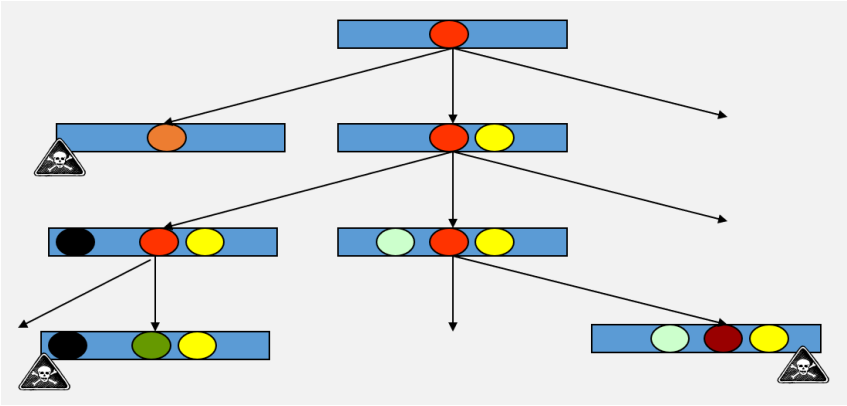
Optimal multiple sequence alignment

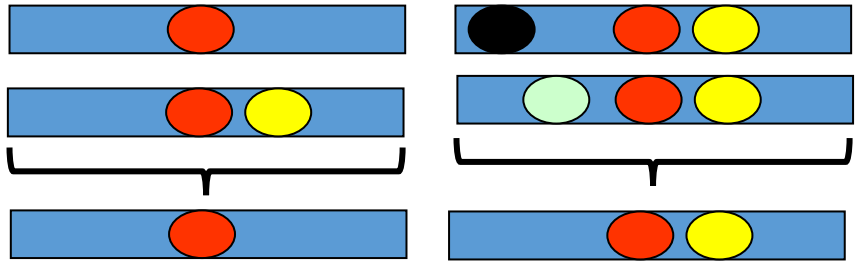
ClustalW: Heuristics-based multiple sequence alignment

Applying multiple sequence alignment

What & why of multiple alignment

Exercise





If a column has the same amino acid in all n rows, is this due to chance or due to biology?

What if n = 2?

What if n = 3?

What if n is > 10?

Basis of multiple sequence alignment

If sequence similarity is weak, pairwise alignment may not identify biologically related conserved positions

Simultaneous comparison of many sequences often allows us to find similarities that pairwise sequence comparison fails to reveal

Bioinformaticians sometimes say that while pairwise alignment whispers, multiple alignment shouts

Multiple sequence alignment maximizes number of positions in agreement across several sequences

Sequences belonging to same “family” usually have more conserved positions in a multiple sequence alignment than sequences not in the same family

```
gi|126467|      FHFTSWPDFGVPFTPIGMLKFLKKVKACNP--QYAGAIVVHCSAGVGRTGTFVVIDAMLD
gi|2499753      FHFTGWPDHGVPYHATGLLSFIRRVKLSNP--PSAGPIVVHCSAGAGRTGCYIVIDIMLD
gi|462550|      YHYTQWPDMGVPEYALPVLTFVRRSSAARM--PETGPVLVHCSAGVGRTGTYIVIDSMLQ
gi|2499751      FHFTSWPDHGVPDTTDLLINFRYLVRDYMKQSPPESPILVHCSAGVGRTGTFIAIDRLIY
gi|1709906      FQFTAWPDHGVPEHPTPFLAFLRRVKTCNP--PDAGPMVVHCSAGVGRTGCFIVIDAMLE
gi|126471|      LHFTSWPDFGVPFTPIGMLKFLKKVKTLNP--VHAGPIVVHCSAGVGRTGTFIVIDAMMA
gi|548626|      FHFTGWPDHGVPYHATGLLSFIRRVKLSNP--PSAGPIVVHCSAGAGRTGCYIVIDIMLD
gi|131570|      FHFTGWPDHGVPYHATGLLGFVRQVKSKSP--PNAGPLVVHCSAGAGRTGCFIVIDIMLD
gi|2144715      FHFTSWPDHGVPDTTDLLINFRYLVRDYMKQSPPESPILVHCSAGVGRTGTFIAIDRLIY
                 ..* *** ***      . *                 ..***** ****... ** ..
```

Conserved sites

Optimal multiple alignment

From pairwise to multiple alignment

Alignment of 2 sequences is represented as a 2-row matrix

Alignment of 3 sequences is represented as a 3-row matrix in a similar way:

A	T	–	G	T	T	a	T	A
A	g	C	G	a	T	C	–	A
A	T	C	G	T	–	C	T	c

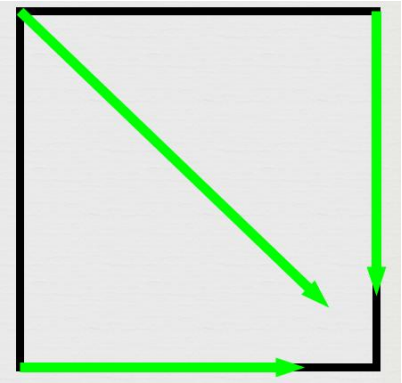
More conserved columns ⇒ the alignment is better

Exercise

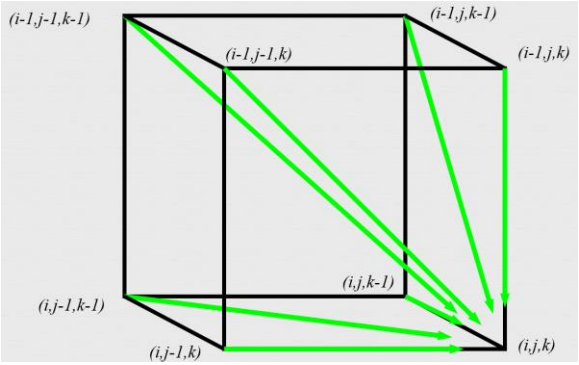
Write down a solution for 3-way optimal global alignment

Hint:

In pairwise alignment, each step in a dynamic programming solution considers 3 possible paths – (mis)match, insert, delete



In a 3-way alignment, each step considers 7 possible paths



Time complexity of optimal multiple alignment

For 3 sequences of length n, the run time is $O(7n^3) = O(n^3)$

For k sequences, the run time is $O((2^k - 1)(n^k)) = O(2^k n^k)$

$\Rightarrow$ Dynamic programming for alignment between two sequences is easily extended to k sequences but it is impractical

$\Rightarrow$ Heuristics-based method, e.g. ClustalW

ClustalW:
Heuristic-based multiple alignment

ClustalW

Widely used program for multiple sequence alignment (MSA) of DNA or protein sequences

ClustalW's progressive alignment strategy

Compute how similar every pair of sequences is

Build a guide tree to determine the order in which sequences should be aligned

Align sequences step by step, following the guide tree

The tree indicates which sequences are most similar and should be aligned first

Once a group is aligned, it is treated as a single entity in later alignments. Weights are applied so that closely related sequences do not dominate the final alignment

Some features that improve alignment quality

Position-specific gap penalties: Gaps are penalized more in conserved regions than in variable ones

Sequence weighting: Reduces bias from redundant sequences

Scoring matrices: Uses PAM or BLOSUM for proteins to reflect evolutionary substitution likelihoods

Step 1 of ClustalW: Pairwise alignment

Aligns each sequence against each other giving a similarity matrix

Similarity = exact matches / sequence length (percent identity)

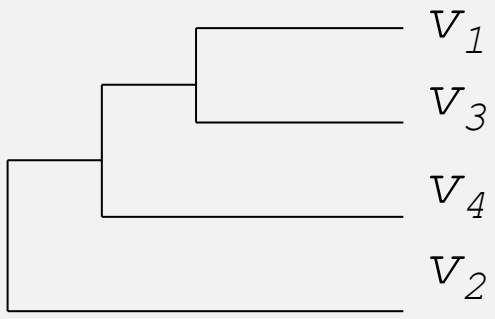
	v_1	v_2	v_3	v_4
v_1	-			
v_2	.17	-		
v_3	.87	.28	-	
v_4	.59	.33	.62	-

(.17 means 17 % identical)

Source: Somayyeh Koohi

Step 2 of ClustalW: Guide tree construction

	v_1	v_2	v_3	v_4
v_1	–			
v_2	.17	–		
v_3	.87	.28	–	
v_4	.59	.33	.62	–



ClustalW uses neighbour-joining to build guide tree

Guide tree roughly reflects evolutionary relations

We will talk more about "neighbour-joining" in a later lecture on phylogenetic trees

Source: Somayyeh Koohi

Step 3 of ClustalW: Tree-based recursion

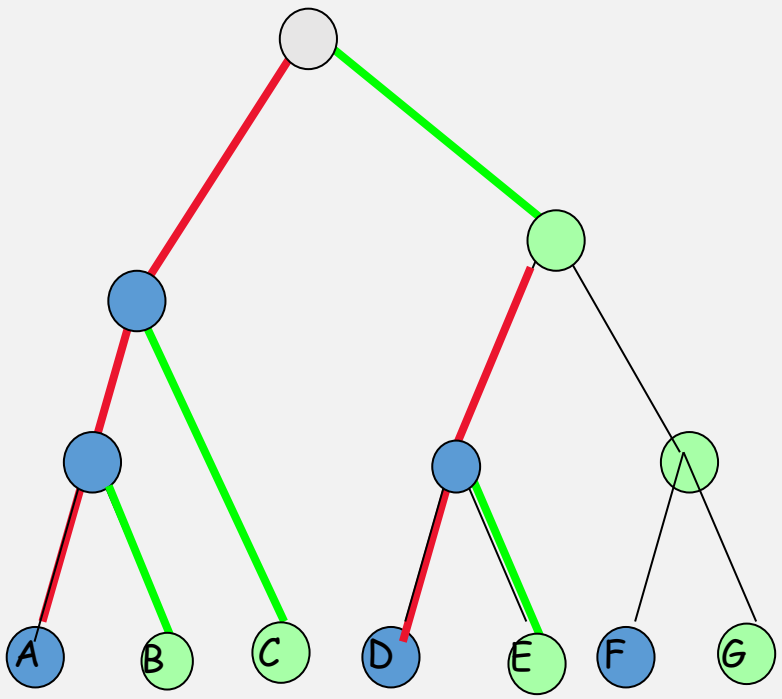
Align(node N) {

Set A_1 = If N's left child is a node
Then Align(N's left child)
Else N's left child

Set A_2 = If N's right child is a node
Then Align(N's right child)
Else N's right child

Return profileAlignment of A_1, A_2

}



Source: Somayyeh Koohi

Exercise

ClustalW progressively aligns sequences following a guide tree

It has to align two groups of sequences in two subtrees at some node

How does ClustalW do this?

Treat each cluster as a profile

Each cluster (group of aligned sequences) is represented as a **profile**, which summarizes:

- The **frequency** of each residue (or nucleotide) at every column in the alignment.
- The **gap frequency** at each column.

Slightly simplified!

For example, if cluster A has 5 sequences and in one column 4 have "A" and 1 has "G", then:

yaml Copy code

```yaml
P_A(column) = {A: 0.8, G: 0.2, others: 0}
```

Profile representation of multiple alignment, simplified

Alignment

T	C	G	G	G	–	g	T	T	T	t	t
c	C	–	–	t	G	A	c	T	T	a	C
a	C	G	–	G	G	A	T	T	T	t	C
T	t	G	G	G	–	A	c	T	T	t	t
a	–	–	–	G	–	–	–	T	–	C	–
T	t	G	G	G	G	A	c	T	T	C	C
T	C	G	–	–	G	A	T	T	c	a	t
–	–	–	G	G	G	A	T	T	c	C	–
T	a	G	G	G	G	A	a	c	–	–	C
T	C	G	G	G	t	A	T	a	a	C	C

Profile

A:	.2	.1	0	0	0	0	.8	.1	.1	.1	.2	0
C:	.1	.5	0	0	0	0	0	.3	.1	.2	.4	.5
G:	0	0	.7	.6	.8	.6	.1	0	0	0	0	0
T:	.6	.2	0	0	.1	.1	0	.5	.8	.6	.2	.3

Source: Somayyeh Koohi

Exercise

Actually, ClustalW employs sequence weighting to reduce bias from redundant sequences when generating profiles

Discuss how this can be done

Compute profile-profile similarity

To align two profiles (say, cluster A and cluster B), ClustalW computes a **score for aligning each column i of A to each column j of B** using:

$$\text{ColumnScore}(i, j) = \sum_x \sum_y P_A(i, x) \times P_B(j, y) \times \text{Score}(x, y)$$

where:

- $P_A(i, x)$ = frequency of residue x at position i in cluster A,
- $P_B(j, y)$ = frequency of residue y at position j in cluster B,
- Score(x, y) = substitution score from a matrix (e.g., BLOSUM62 or PAM).

This effectively measures how well two columns of residues (rather than two residues) match.

Profile-profile alignment by dynamic programming

Once all column–column scores are computed, ClustalW applies **global dynamic programming** (Needleman–Wunsch–type algorithm) to align the two profiles:

- The match/mismatch score is the profile–profile similarity ColumnScore(i, j)
- The gap penalties are adjusted based on:
  - Whether the region is conserved (higher penalty for gaps in conserved regions).
  - The average gap frequencies in each profile (lower penalty in variable regions).

what the dynamic programming looks like

```
# Initialize DP borders
DP[0][0] = 0
for i in 1..m: DP[i][0] = DP[i-1][0] - GAP_PENALTY(ProfileA, i)
for j in 1..n: DP[0][j] = DP[0][j-1] - GAP_PENALTY(ProfileB, j)

# Fill DP matrix
for i in 1..m:
    for j in 1..n:
        match = DP[i-1][j-1] + COLUMN_SCORE(PA[i], PB[j])
        delete = DP[i-1][j] - GAP_PENALTY(ProfileA, i)
        insert = DP[i][j-1] - GAP_PENALTY(ProfileB, j)
        DP[i][j] = max(match, delete, insert)
    end for
end for
```

ClustalW adjusts gap penalties dynamically

Gaps in conserved regions → higher penalty

Gaps near existing gaps or variable regions → lower penalty

```
function GAP_PENALTY(Profile, position)
    base_penalty = g_open + g_extend
    if Profile[position] is highly conserved:
        return base_penalty * 1.5      # discourage gaps
    else if near existing gap:
        return base_penalty * 0.5      # easier to open gaps
    else:
        return base_penalty
end function
```

Exercise

How does ClustalW tell whether a profile position is highly conserved?

```
function GAP_PENALTY(Profile, position)
    base_penalty = g_open + g_extend
    if Profile[position] is highly conserved:
        return base_penalty * 1.5      # discourage gaps
    else if near existing gap:
        return base_penalty * 0.5      # easier to open gaps
    else:
        return base_penalty
end function
```

Applying multiple alignment

Protein domains and motifs

Domains are…

Large self-stabilizing units that fold independently in a protein

Shared across proteins from different genes

Crucial for protein function

Can be swapped between proteins to create chimeras

May consist of one or several structural motifs; some domains do not correspond neatly to any single motif

Motifs are …

Smaller recurring structural patterns (like helix–turn–helix or zinc finger)

Discovering domain and active sites

```
>gi|475902|emb|CAA83657.1| protein-tyrosine-phosphatase alpha
MDLWFFVLLLGSGLISVGATNVTTEPPTTVPTSTRIPTKAPTAAPDGGTTPRVSSLNVSSPMTTSAPASE
PPTTTATSISPNATTASLNASTPGTSVPTSAPVAISLPPSATPSALLTALPSTEAEMTERNVSATVTTQE
TSSASHNGNSDRRDETPIIAVMVALSSLLVIVFIIIVLYMLRFKKYKQAGSHSNSFRLPNGRTDDAEPQS
MPLLARSPSTNRKYPPLPVDKLEEEINRRIGDDNKLFREEFNALPACPIQATCEAASKEENKEKNRYVNI
LPYDHSRVHLTPVEGVPDSHYINTSFINSYQEKNKFIAAQGPKEETVNDFWRMIWEQNTATIVMVTNLKE
RKECKCAQYWPDQGCWTYGNIRVSVEDVTVLVDYTVRKFCIQQVGDVTNKKPQRLVTQFHFTSWPDFGVP
FTPIGMLKFLKKVKTCNPQYAGAIVVHCSAGVGRTGTFIVIDAMLDMMHAERKVDVYGFVSRIRAQRCQM
VQTDMQYVFIYQALLEHYLYGDTELEVTSLEIHLQKIYNKVPGTSSNGLEEEFKKLTSIKIQNDKMRTGN
LPANMKKNRVLQIIPYEFNRVIIPVKRGEENTDYVNASFIDGYRRRTPTCQPRPVQHTIEDFWRMIWEWK
SCSIVMLTELEERGQEKCAQYWPSDGSVSYGDINVELKKEEECESYTVRDLLVTNTRENKSRQIRQFHFH
GWPEVGIPSDGKGMINIIAAVQKQQQQSGNHPMHCHCSAGAGRTGTFCALSTVLERVKAEGILDVFQTVK
SLRLQRPHMVQTLEQYEFCYKVVQEYIDAFSDYANFK
```

How do we find the domain and associated active sites in the protein above?

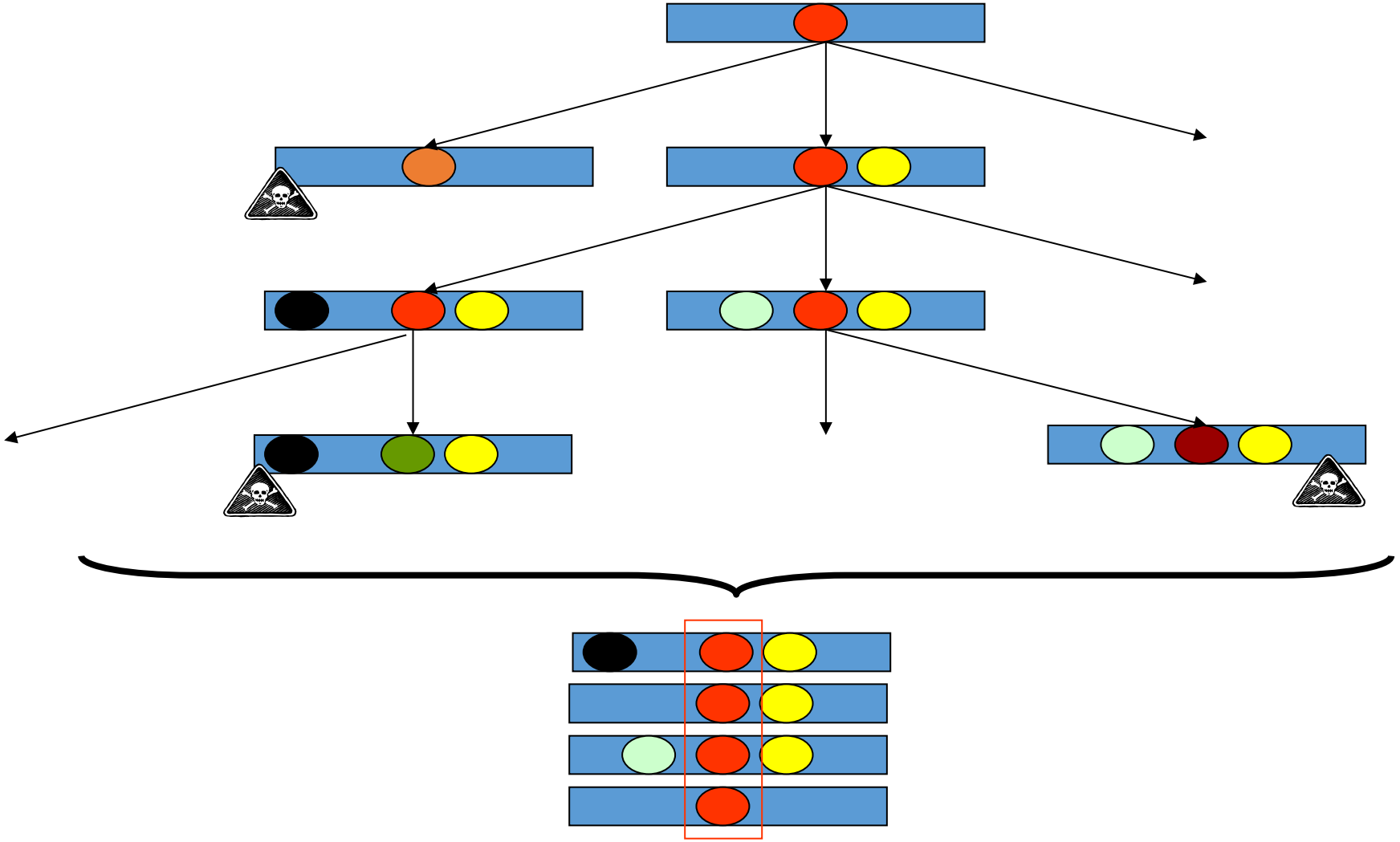
Domain/active sites as emerging patterns

How to discover active site and/or domain?

If you are lucky, domain has already been modelled
BLAST, HMMPFAM, …

If you are unlucky, domain not yet modelled
Find homologous seqs
Do multiple alignment of homologous seqs
Determine conserved positions
$\Rightarrow$ *Emerging patterns relative to background*
$\Rightarrow$ *Candidate active sites and/or domains*

In the course of evolution…



Multiple sequence alignment maximizes number of positions in agreement across several sequences

Sequences belonging to same “family” usually have more conserved positions in a multiple sequence alignment than sequences not in the same family

```
gi|126467|      FHFTSWPDFGVPFTPIGMLKFLKKVKACNP--QYAGAIVVHCSAGVGRTGTFVVIDAMLD
gi|2499753      FHFTGWPDHGVPYHATGLLSFIRRVKLSNP--PSAGPIVVHCSAGAGRTGCYIVIDIMLD
gi|462550|      YHYTQWPDMGVPEYALPVLTFVRRSSAARM--PETGPVLVHCSAGVGRTGTYIVIDSMLQ
gi|2499751      FHFTSWPDHGVPDTTDLLINFRYLVRDYMKQSPPESPILVHCSAGVGRTGTFIAIDRLIY
gi|1709906      FQFTAWPDHGVPEHPTPFLAFLRRVKTCNP--PDAGPMVVHCSAGVGRTGCFIVIDAMLE
gi|126471|      LHFTSWPDFGVPFTPIGMLKFLKKVKTLNP--VHAGPIVVHCSAGVGRTGTFIVIDAMMA
gi|548626|      FHFTGWPDHGVPYHATGLLSFIRRVKLSNP--PSAGPIVVHCSAGAGRTGCYIVIDIMLD
gi|131570|      FHFTGWPDHGVPYHATGLLGFVRQVKSKSP--PNAGPLVVHCSAGAGRTGCFIVIDIMLD
gi|2144715      FHFTSWPDHGVPDTTDLLINFRYLVRDYMKQSPPESPILVHCSAGVGRTGTFIAIDRLIY
                 ..* *** ***      . *                 ..***** ****... ** ..
```

Conserved sites

Exercise

Some protein tyrosine phosphatases (PTP) have 2 PTP domains

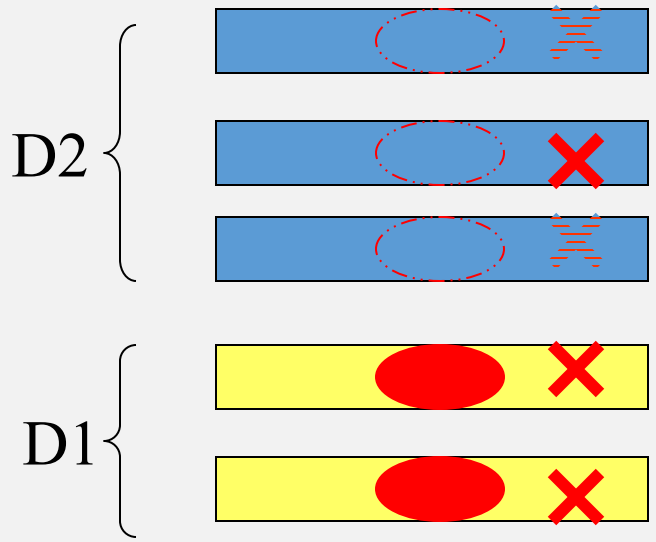
PTP domain D1 has much more activity than PTP domain D2

How do to figure that out which mutations are responsible for this difference?

Sequence from a typical PTP domain D2

```
>gi|00000|PTPA-D2
EEEFKKLTSIKIQNDKMRTGNLPANMKKNRVLQIIPYEFNRVIIPVKRGEENTDYVNASF
IDGYRQKDSYIASQGPLLHTIEDFWRMIWEWKSCSIVMLTELEERGQEKCAQYWPSDGLV
SYGDITVELKKEEECESYTVRDLLVTNTRENKSRQIRQFHFHGWPEVGIPSDGKGMISII
AAVQKQQQQSGNHPITVHCSAGAGRTGTFCALSTVLERVKAEGILDVFQTVKSLRLQRPH
MVQTLEQYEFCYKVVQEYIDAFSDYANFK
```

Hint: Emerging patterns of PTP D1 vs D2



Which of these two sites (“X” or “O”) is more likely to explain the difference of D1 and D2?

About the inventor: Prasanna Kolatkar

Prasanna Kolatkar

Research Fellow, BIC, NUS, 1997-1999

Currently Senior Scientist at Qatar Biomedical Research Institute

Exercise

What have we learned in this lecture?

Good to read

W. K. Sung. “Multiple sequence alignment”, Chapter 6, *Algorithms in Bioinformatics: A Practical Introduction*. Chapman and Hall, 2009

K.L.Lim et al. “Interconversion of kinetic identities of the tandem catalytic domains of receptor-like protein tyrosine phosphatase PTP-alpha by two point mutations is synergist and substrate dependent”, *JBC*, 273:28986--28993, 1998