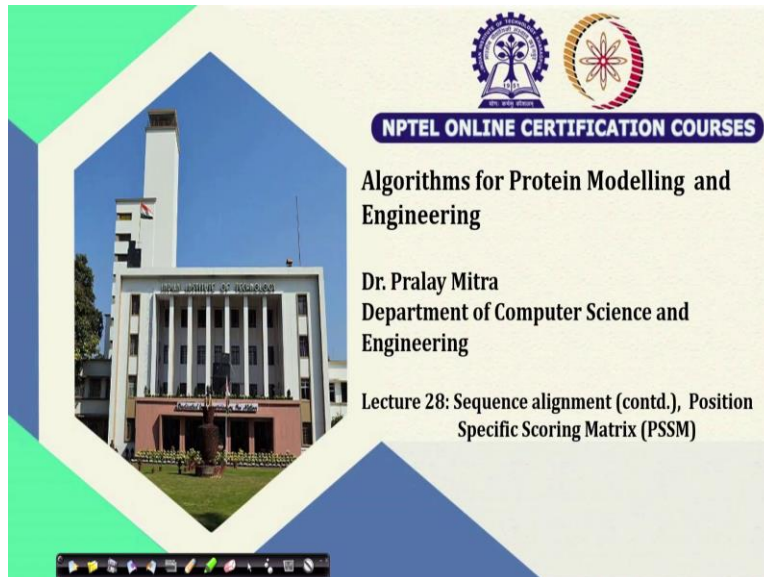


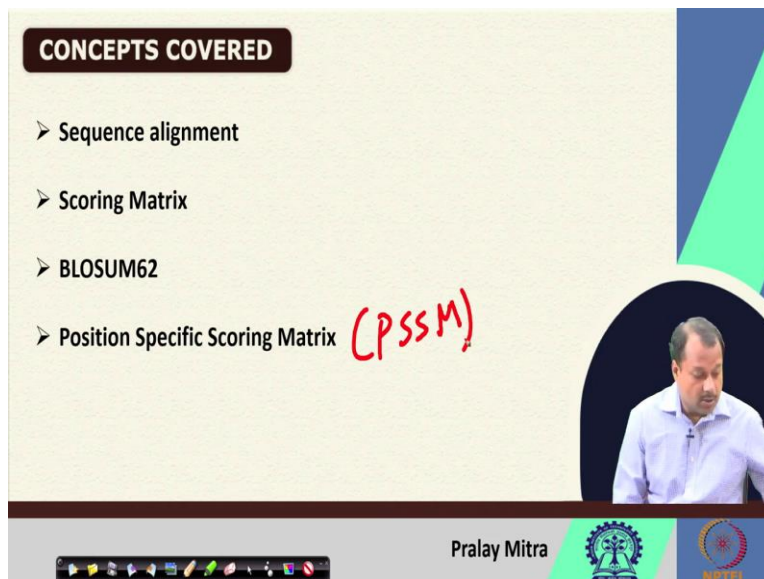
Algorithms for Protein Modelling and Engineering
Professor Pralay Mitra
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur
Lecture 28

Dynamic Programing (Continued), Position Specific Scoring Matrix (PSSM)
(Refer Slide Time: 00:22)



The slide features a central image of a building through a hexagonal frame. To the right, it displays the NPTEL logo and the text 'NPTEL ONLINE CERTIFICATION COURSES'. Below this, the course title 'Algorithms for Protein Modelling and Engineering' is shown, followed by the instructor's name 'Dr. Pralay Mitra' and his affiliation 'Department of Computer Science and Engineering'. The lecture title 'Lecture 28: Sequence alignment (contd.), Position Specific Scoring Matrix (PSSM)' is at the bottom right. A navigation bar is visible at the bottom left.

(Refer Slide Time: 00:29)



The slide is titled 'CONCEPTS COVERED' in a dark box. It lists four topics: 'Sequence alignment', 'Scoring Matrix', 'BLOSUM62', and 'Position Specific Scoring Matrix (PSSM)'. The text '(PSSM)' is written in red handwritten font next to the last item. A small video inset of the professor is in the bottom right corner. The slide footer includes the name 'Pralay Mitra' and the NPTEL logo. A navigation bar is at the bottom left.

(Refer Slide Time: 01:24)

KEYWORDS

- Sequence alignment
- BLOSUM62

Pralay Mitra

IIT Bombay NPTEL

Welcome back. So, let us continue with the sequence alignment problem that we started from the dynamic programming and then continue with the sequence alignment and next we will discuss several scoring function, BLOSUM 62 and then position specific scoring matrix that we will discuss.

So, the concept we are planning to discuss in this lecture is sequence alignment, scoring matrix and BLOSUM 62, position specific scoring matrix in short that is called as the PSSM, so it is called as that PSSM. So, in the context of the sequence alignment, we are discussing the dynamic programming and when it is say dynamic programming.

So, two things we can actually customize or modified there one is the score function, I mean the inside the score function that S the match mismatch score along with whether the gap will be introduced or not and if gap is introduced what kind of penalty will be there, so those things we can discuss. Now, let us continue with that one, that is why I choose the keyword also same like this sequence alignment and BLOSUM 62 matrix, additionally the PSSM will also come.

(Refer Slide Time: 01:29)

Dynamic Programming

Algorithm 10:
Input: Strings **A** and **B** with n and m characters, respectively
Output: The optimal alignment $MAlign[i,j]$ of a longest string that is a subsequence of both the string $A[0..i]$ and the string $B[0..j]$

Steps:

- for $i=1$ to $n-1$ do
 $Malign[i-1] = 0$
- for $j=0$ to $m-1$ do
 $Malign[-1,j] = 0$

Initialization

Diagram: A vertical line with 'n' at the top, '0' at the bottom, and a downward arrow. To its right, a vertical line with 'm' at the top and '0' at the bottom. A diagonal line separates the two, with 'Initialization' written in red.

Pralay Mitra

So, this is our 10th algorithm, which says input string A and B with n and m characters respectively, the output is the optimum alignment M align i,j of a longest string that is a subsequence of both the string A 0 through i, and the string B 0 through i. Now the steps, so this we can call as the initialization step where what we are doing i minus 1 is 0, so for all the i's from 1 through n minus 1.

Optionally as I mentioned what you can consider instead of this minus i, you can make 0 and instead in that case then instead of n minus 1 you can keep it n. That is same for here also, that you can consider as m and then it can, you can consider as 0, that is your implementation. But from algorithm we can also mention that at a initialization phase one column and one row will be appended or augmented, so that is at the beginning. Next will come as you remember the matrix fill or the score function calculation.

(Refer Slide Time: 02:50)

Dynamic Programming

```
3. for i=0 to n-1 do
  1. for j=0 to m-1 do
    if  $a_i = b_j$  then
       $Malign[i, j] = Malign[i-1, j-1] + 1$ 
    else
       $Malign[i, j] = \max\{Malign[i-1, j], Malign[i, j-1], Malign[i-1, j-1] + 1\}$ 
  4. Trace back to retrieve alignment.
```

Scoring

Match = 1

Mismatch = 0

Backtracking

S[i,j]

Pralay Mitra

So, here what we will do that we have to process the matrix so for i equals to 0 through n minus 1, do j equals to 0 through m minus 1 do, if a_i equals to b_j then $Malign$ equals to $Malign$ i minus 1, j minus 1, plus 1. Now, here we did the modification it is going to be my $S_{i,j}$, but from dynamic programming algorithm point of view it is correct.

So, 1 because I am assuming match equals to 1 match and mismatch that is why it is 1, but in a generic way or for the generalization you can write $S_{i,j}$ equals plus $Malign$ i minus 1 j minus 1. But if they will not match, then what you can do, $Malign$ i, j max $Malign$ i, j minus 1 or $Malign$. So, here actually this S will be 0.

So, that way this will be actually removed, so i minus 1 j minus 1, but if you add $S_{i,j}$ then combining these two what you can write that $Malign$ i minus 1 j , $Malign$ i, j minus 1, $Malign$ i minus 1, j minus 1 and here actually will be $S_{i,j}$. And finally, so this is my scoring or matrix fill and this is my traceback or backtracking.

But you remember what I have mentioned that during the scoring or matrix filling, if you keep an information like if you keep an information like from where it has come, so either this or this or this, then during this backtracking or trace back you just reverse this arrow, then the backtracking will be either this or this or this, then it will be easy for you.

Now, regarding this arrow, I understand that you cannot able to store that arrow, do not look for the special character where there is an arrow that is no need of it. So, you have in your

implementation one concept that say for example, you can consider that this is going to be my say 0, this is going to be 1 and this is going to be 2. So, this is your convention in your implementation that if it is 0, I will go diagonally I will go directly up, if it is 1, I will go diagonally up, if it is 2, I will go to the left. So, 0, 1, 2, three states used here.

(Refer Slide Time: 05:44)

Dynamic Programming

Time complexity: $O(n \times m)$
where n and m are the lengths of the strings

```
3. for i=0 to n-1 do
  1. for j=0 to m-1 do
    if  $a_i = b_j$  then
       $Malign[i, j] = Malign[i-1, j-1] + 1$ 
    else
       $Malign[i, j] = \max\{Malign[i-1, j], Malign[i, j-1], Malign[i-1, j-1] + S\}$ 
  4. Trace back to retrieve alignment.
```

Implementation?

Pralay Mitra

The time complexity as we discussed is n cross m , where n and m is the length of two strings which is supplied and you are going for at pairwise alignment, this is going to be your space complexity also, because you need to declare or define one matrix of the same size n cross m actually it will be $n + 1$ cross $m + 1$, but that plus 1 you know that during the complexity calculation it will not come that is why it will be n , m or n square assuming both the strings are of same length.

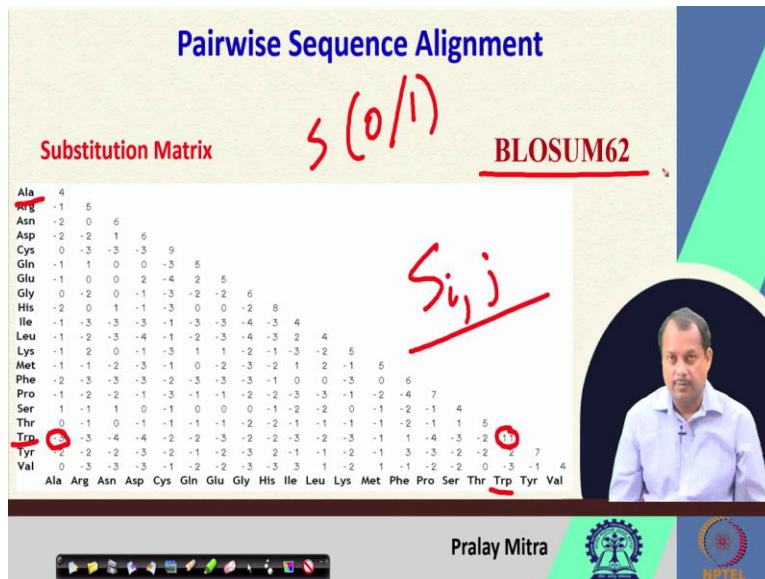
But remember, I said same length, but they not may not be the same one. But even they are order is same you can also consider in square, but ideally it is n cross m or m , n for space and for space as well as the time. So, I believe now it will be easy for you to go for the implementation, the gap penalty that we have discussed on the last lecture and the detail algorithm is here, so you just need to code each line which is kind of a pseudo code and that is it.

So, the data structure you need to have is your one matrix that I mentioned is given here, so better is that after reading the string, you dynamically allocate the memory location, which will

be required to you. You can go for static allocation, if you have a previous idea that what could be the maximum length of your string then with that you can go.

But if you go for the static allocation of the memory, then my suggestion will be after reading the string you please check whether your allocated memory is sufficient enough to support your, to support your input string or not if yes, go ahead, if not then perhaps you have to exit from the program you have to change your source code and run it again. So, that is why if you go for dynamic memory allocation like C used to do and other languages also use to support that one it will be better.

(Refer Slide Time: 07:46)



Now, here comes for which we are waited long, that is $S(i,j)$, of the score matrix. Now, here you can see along the row and column 20 different amino acids are there, they are sorted by their, they are alpha lexicographically sorted by their name, so that is why alanine, arginine, asparagine aspartic acid, cysteine, glutamine then glutamic acid then glycine, histidine, isoleucine, leucine, lysine, methionine, then phenelzine, proline, serine, threonine, tryptophan tyrosine and valine. So, those are coming that way.

And you can see that there is an integer matrix as I promise that this matrix calculation in this is going to be an integer matrix. Now here you can see that there are negative values, positive values, zeroes all are there. So positive values mostly with the highest value is at the diagonal

position when say alanine will be substituted by alanine and arginine will be substituted by arginine like that way.

And that way you see that even when the highest number is occurring for tryptophan is as for which, tryptophan is being yes tryptophan is being replaced by tryptophan. But as I mentioned that when say you will go for tryptophan is being say tryptophan is being replaced by alanine then it is minus 3. So, it is not only, I am not allowing that one, but I am again in some sort of I am again giving some sort of penalty here, so, minus 3.

And all those minus values you can consider is kind of a penalty. So, which indicates that it is not preferable, so, you should not go with this. And that way you can see that which was very simple for you as S binary values 0 or 1 is now having a lot of variations and that variations are following some evolution. So, this substitution matrix is called as the BLOSUM 62, why it is BLOSUM? From where it has come? I will go to the next slide.

But before that, I wish to say that here I am considering that the operation is symmetric in nature, I mean that during that evolution process the mutation will be symmetric. So, the probability of mutating alanine by say, arginine is same as mutating arginine by alanine, that is why I am taking only the lower triangular matrix.

But there is a variation of these BLOSUM 62 metrics also, where 20 cross 20 I mean, upper triangular matrices are also there, where the variations in the values are observed theoretically speaking that in the evolution process mutating alanine by arginine, and vice versa is not same, they have some minor differences.

But what is what people have noted that, if you consider say 20 cross 20 matrix or this matrix, so from sequence alignment point of view, you did not get much difference, so that is why a probably it is better to stick on with the one triangular matrix, I mean the lower triangular matrix with the diagonal. So, that is why people have considered this. So, this is your first substitution matrix, there may be a number of other substitution matrix but this BLOSUM 62 is most widely used one. So, what is BLOSUM 62?

(Refer Slide Time: 11:15)


(BLOcks SUbstitution Matrix)

$$S_{ij} = \left(\frac{1}{\lambda}\right) \log\left(\frac{p_{ij}}{q_i \times q_j}\right)$$


p_{ij} is the probability of two amino acids i and j replacing each other in a homologous sequence,

q_i and q_j are the background probabilities of finding the amino acids i and j in any protein sequence,

λ is a scaling factor.



Pralay Mitra



(BLOcks SUbstitution Matrix)


$$S_{ij} = \left(\frac{1}{\lambda}\right) \log\left(\frac{p_{ij}}{q_i \times q_j}\right)$$

p_{ij} is the probability of two amino acids i and j replacing each other in a homologous sequence,


q_i and q_j are the background probabilities of finding the amino acids i and j in any protein sequence,

λ is a scaling factor.

BLOSUM 62



Pralay Mitra



The name BLOSUM has come from blocks substitution matrix, and how it is being computed? S_{ij} again it is in the form of a matrix, so i, j indicates two amino acid positions, as you see on the last slide only that alanine arginine aspartic acid like that way 20 amino acids are there. So, if I say i, j which means a alanine one amino acid, two another amino acid, so that probability is one upon one divided by lambda log of P_{ij} divided by q_i multiplied with q_j .

So, sorry about the typo. So, this j this j is going to be the subscript, so it will be q_j , sorry about the typo, but it is correct here. Now, what is P_{ij} ? P_{ij} is the probability of two amino acids i and j replacing each other in a homologous sequence. So, what is homology? I will go to that one. So,

for the time being you can consider that when they have, they are mostly same then they are called as a homologous. But that is not the correct definition. Actual definition I will give you later. But for the time being you can consider.

Now, q_i and q_j are the background probabilities of finding the amino acids i and j in any protein sequence. So, you are given with a number of protein sequences. Now, from that protein sequences, you pick two amino acids without any loss of generality, let us assume arginine sorry you have picked arginine and say cysteine, sorry cysteine, so this is R and this is C you have picked these two amino acids.

Now, given a database of protein sequences, you check what is the probability of occurrence of arginine, what is the probability of occurrence of the cysteine, that is your q_i and q_j . And from that database you compute the homologous sequences, when you compute the homologous sequences, which are homologous in nature from there you check that in, what is the probability of replacing arginine by cysteine, you have that one as your P_{ij} .

If you have that one then with respect to thier data set you can compute your S_{ij} . Now I considered arginine and cysteine for 20 different i and 20 different j you can have that and then you will get 20 cross 20 matrix. Now, it is up to you whether you will consider one triangular matrix along with the diagonal or all 20 cross 20, but that is the way you can consider, you can compute.

Now, λ is a scaling factor you can scale it accordingly because as you understand that this is a log operation probability values and then it is going to be the fractional one and when it is fractional one, then you know that in computer so floating point arithmetic is more time consuming compared to the integer arithmetic.

So, it might be a good idea that you convert your floating-point number after some scaling to some integer value, because just now we have seen and also, I will show you again here you see all the values are integer in nature. And if they are integer in nature, definitely the matrix, the dynamic programming is also going to be integer in nature, so we will be using integer arithmetic operation that is faster compared to the floating-point number.

So, that is why it is better to actually avoid floating point number that is why I am using some scaling factor. Then comes one thing that is on the last slide I mention BLOSUM 62, so this is

BLOSUM, so the name you can understand from here, so this M is coming from here, this U is from here, and BLO from here, what is 62? Remember just now, I mentioned that you have one database, so from that database you are calculating the q_i and q_j corresponding to each amino acids that is a probability of their occurrences in the background.

Then you computed P_{ij} which is the probability of replacing one amino acid i th amino acid by the j th amino acid and when you are doing then you are doing or performing this one from a data set. Now, when there is a data set then what is the similarity among the sequences in that data set that will matter. So, that 62 indicates the percentage of similarity inside that sequences, in that database.

So, that way they are maybe 80, BLOSUM 80, BLOSUM 30, BLOSUM 40, BLOSUM 50 indicating what is the maximum similarity achieved there. But it has been tested and people has understand that BLOSUM 62 gives the best performance compared to the others that is why BLOSUM 62 is going to be one standard in our case.

(Refer Slide Time: 16:40)

(BLOcks SUBstitution Matrix)

- BLOSUM matrices with high numbers are designed for comparing closely related sequences, while those with low numbers are designed for comparing distant related sequences.
- BLOSUM 80 is used for less divergent alignments, and BLOSUM 45 is used for more divergent alignments.

Pralay Mitra

The slide features a video inset of Pralay Mitra in the bottom right corner. The background is light yellow with a blue and green geometric design on the right side. The title is in blue and red text. The bullet points are in black text. The video inset shows a man in a light blue shirt speaking.

So, BLOSUM matrices with high numbers are designed for comparing closely related sequences, high number means, the sequence similarity is very high, while those with low numbers are designed for comparing distant related sequences. So, that way if you know the equation just now, I have shown you the equation using that equation from a data set actually you can compute your own BLOSUM matrix.

So, BLOSUM 80 is used for less divergent alignments and BLOSUM 45 is used for more divergent alignment. So, these are two examples that you can see 80 and 45. So, that 80 indicates that less divergent alignments and 45 indicates that more divergent or the variety is more in the sequences.

(Refer Slide Time: 17:33)

(BLOcks SUbstitution Matrix)

- The matrices were created by merging (clustering) all sequences that were more similar than a given percentage into one single sequence and then comparing those sequences (that were all more divergent than the given percentage value) only; thus reducing the contribution of closely related sequences.
- The percentage used was appended to the name, giving BLOSUM80 for example where sequences that were more than 80% identical were clustered.

BLOSUM62

Pralay Mitra

IITM NPTEL

The matrices were created by marching all sequences that were more similar than a given percentage into one single sequence and then comparing those sequences that were all more divergent than a given percentage value only thus reducing the contribution of closely related sequences, the percentage used was appended to the name giving BLOSUM 80 for example of our sequences that were more than 80 percent identical were clustered.

So, now you understand the meaning properly. So, when I say now, now you can able to understand and answer me if I asked you that when I say BLOSUM 62, so what that 62 signifies? It says that more than 62 percent are identical sequences are clustered. Now, if it is BLOSUM 80 which means more similar sequences are clustered, so I will get the similarity more, if it is 45 then 45 percent and more, sequence similar sequences are clustered, so I will get more divergent information. So, this we discussed.

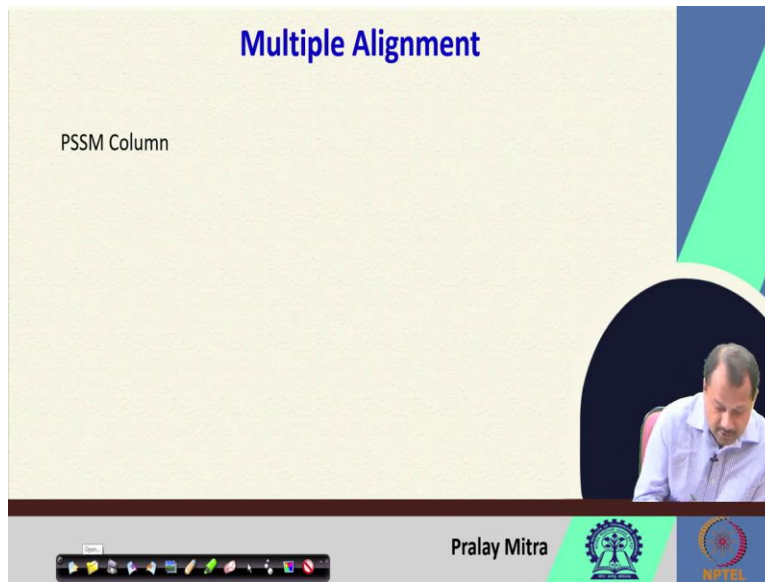
(Refer Slide Time: 18:52)

The slide is titled "Other variations of scoring matrices" in blue text. It features a bulleted list on the left side with three items: "Position Specific Scoring Matrix", "Structural information", and "3D-1D mapping". On the right side, there is a circular video inset showing a man in a light blue shirt, identified as Pralay Mitra. At the bottom of the slide, there is a navigation bar with a toolbar on the left, the name "Pralay Mitra" in the center, and two logos on the right: the Indian Institute of Technology (IIT) logo and the NPTEL logo.

Now, there are some other variations of scoring matrices, so one is called as the position specific scoring matrix, another is structural information based, another is the 3D to 1D mapping. So, this 3D to 1D mapping was done long back by David Rosenberg from the University of California Los Angeles. So, what he has done that when protein structure is given, then you know that structural alignment getting the information from the structure and using that one is bit complicated.

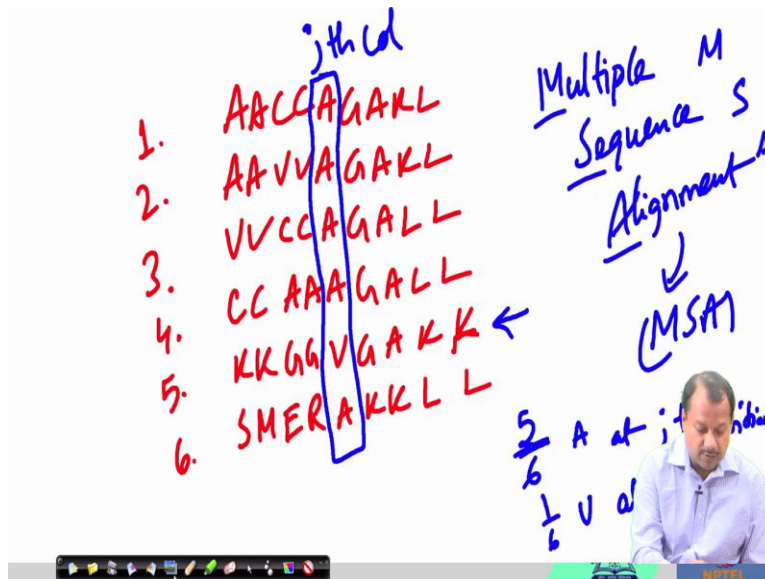
So, why do not we convert that 3D structure information in some way as 1D information when it is in 1D information, I can consider that as a sequence I use the dynamic programming just now I discussed and then I can infer something regarding the similarity or whether they are matching et cetera from there, so why not. But we are not interested about the second and third, but we will discuss the first one that is position specific scoring matrix in short PSSM.

(Refer Slide Time: 19:46)



So, PSSM starts with a multiple alignment, although explicitly we do not discuss the multiple alignment but you understand that when we know that pairwise alignment then given n number of sequences, we can go for multiple alignment using the dynamic programming or other techniques are also there.

(Refer Slide Time: 20:19)



Now, let us assume that there is some multiple alignment and we can also assume that they are the multiple alignment sequences are say AACCAGAKL. Let us assume there are 6 such

sequences some random sequences I considered say only a part of it. Now, if I consider, so these are multiple sequence alignment, or in short that is called as the MSA.

Now, I consider one column, in this column what I see is mostly it is A in one instance only I am getting a V, and that is at this sequence. Then what I can compute? I can compute, so if I assume that this is my j th column then what I can say that, so how many number of occurrences 6, 6 number of occurrences, so 5 by 6 is the probability of occurring A at j th position and 1 by 6 is probability of occurrence of the V at the j th position.

Now, if these two probability values, if these two probability values are with me, then using these two probability values I can actually calculate that what will be the score value when say I am doing some sequence alignment. So, again to that scoring matrix one was the BLOSUM 62 or you can, we also discussed that you can have say BLOSUM 80, BLOSUM 45, so those things you can do.

Now, what I am proposing is that instead of these say that scoring function, you can have another scoring function from the multiple sequence alignment, and that is the position specific scoring matrix which says that at the i th position, so the probability of having A is say 5 by 6 and probability of having V is 1 by 6. So, this PSSM actually has a lot of applications in different areas like protein design, so the definition I am giving to you at this position, but the detail application we will discuss when we will discuss the protein design algorithm.

(Refer Slide Time: 23:50)

Position-based Sequence Weights

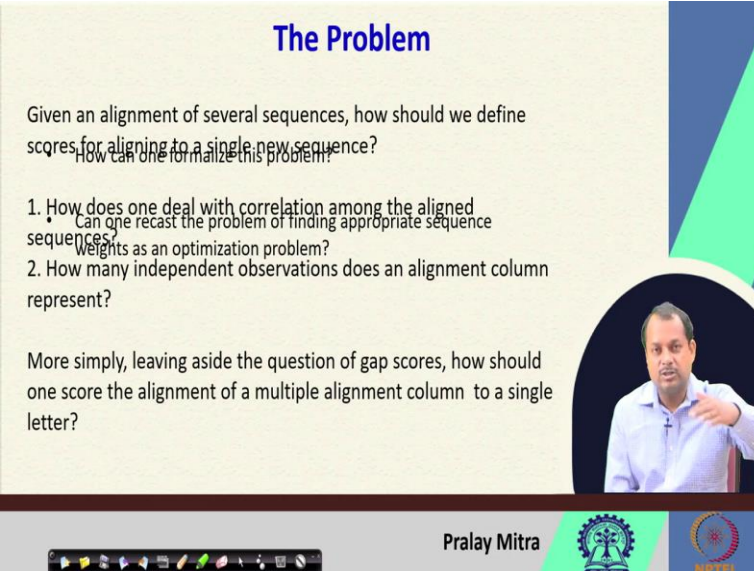
- Use to
 - reduce redundancy
 - emphasize diversity
- Based on
 - distance between a sequence and an ancestral or generalized sequence
 - weights on the diversity observed at each position in the alignment
- Application is in
 - MSA
 - Sequence searching

Pralay Mitra

NPTEL

Now, based upon this position specific scoring matrix, so some sequence may have some weight value and that weight value is used to reduce redundancy to emphasize the diversity and it is based on distance between a sequence and an ancestral or generalized sequence, weights on that diversity observed at each position in the alignment, and it has an application in MSA which means multiple sequence alignment or sequence searching. So, from here we move on to one situation.

(Refer Slide Time: 24:32)



The Problem

Given an alignment of several sequences, how should we define scores for aligning to a single new sequence?
How can one formalize this problem?

1. How does one deal with correlation among the aligned sequences?
Can one recast the problem of finding appropriate sequence weights as an optimization problem?
2. How many independent observations does an alignment column represent?

More simply, leaving aside the question of gap scores, how should one score the alignment of a multiple alignment column to a single letter?

Pralay Mitra

The slide features a video inset of a man in a light blue shirt speaking. At the bottom, there is a navigation bar with icons and logos for IIT Bombay and NPTEL.

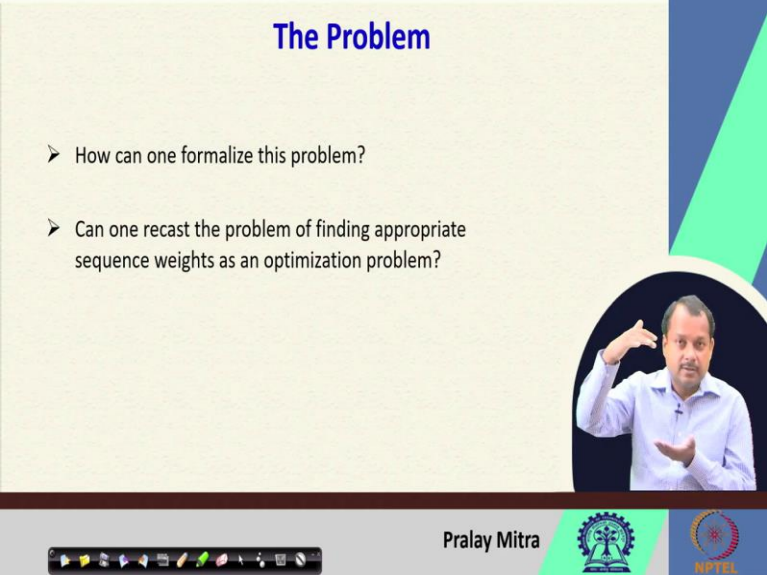
So, I am sorry for this typo again here. But here given an alignment of several sequences, how should we define scores for aligning to a single new sequences, that is important. The reason is that when say you are trying to align too many sequences, then if it is the case that similar sequences are coming more frequently then it will bias towards that one, or if diverse sequences are coming, then it will be very difficult or it will be hard to identify what is the similarity between those.

So, to make a trade-off between these, so given an alignment of several sequences that is already there, now in new sequences has come, so what to fit in there? So, that score value that we wish to compute. So, how does one deal with correlation among the aligned sequences? How many independent observations does an element column represent? So, those things we need to discuss. So, that is the problem.

Now, more simply leaving aside the question of gap scores that we have discussed gap penalty and the score values, how should one score the alignment of a multiple alignment column to a single later? That is our main interest. So, just now in the PSSM matrix, I mentioned that the i th column I see out of 6 number of times, so 6 number of sequences are there, out of, out of that 5 alanine occurs and once valine occurs for one sequence. So, it is the 5 by 6 probability of occurring alanine and 1 by 6 probability of occurring valine.

Now, if a new sequence will come, and for that, at the j th position if we see that it is say valine, which means that if I give reward to that new sequence for the alignment, or placing that one in the previous alignment, then actually I am looking for more diversity because I am favoring 1 by 6 over 5 by 6. On the other hand, if I, if I reduce its weight in that alignment, then basically I am favoring the alanine which is 5 by 6 which means I am favoring which is most occurring, so that cases. So, what will be the trade off and what is the advantage or disadvantage that of that.

(Refer Slide Time: 27:01)



The slide is titled "The Problem" in blue text. It contains two bullet points:

- How can one formalize this problem?
- Can one recast the problem of finding appropriate sequence weights as an optimization problem?

In the bottom right corner, there is a circular video inset showing a man in a light blue shirt gesturing with his hands. Below the slide, there is a navigation bar with a toolbar on the left, the name "Pralay Mitra" in the center, and logos for IIT Bombay and NPTEL on the right.

So, how can one formalize this problem? Can one recast the problem of finding appropriate sequence weights as an optimization problem, which means once one sequence will come, then I will try to calculate one weight value corresponding to that sequence so that using that weight value, it will be actually placed using its appropriate order, or weight, that I am going to do.

(Refer Slide Time: 27:30)

Orthology and Paralogy

Homology: Two genes or proteins are homologous if they share a common ancestor.

Orthology: Two genes or proteins are orthologous if they diverged by speciation.

Paralogy: Two genes or proteins are paralogous if they diverged by gene duplication.

Pralay Mitra

The slide features a title 'Orthology and Paralogy' in blue. Below it are three definitions: 'Homology: Two genes or proteins are homologous if they share a common ancestor.', 'Orthology: Two genes or proteins are orthologous if they diverged by speciation.', and 'Paralogy: Two genes or proteins are paralogous if they diverged by gene duplication.' A video inset shows a man in a light blue shirt. At the bottom, there is a navigation bar with icons, the name 'Pralay Mitra', and logos for IIT Kharagpur and NPTEL.

From there the concept of orthology, paraology comes. So, homology two genes or proteins are homologous, if they share a common ancestor, that is the definition of the homology. Now, you know, in more detail regarding the definition of a homologous sequences in the context of biology that I used a few slides ago.

So homologous then means that two genes or proteins will be homologous if they share a common ancestor. Now, in this homology, there are two terms one is the orthology and other is a paralogy, orthology indicates two genes or proteins are orthologous if they diverged by gene speciation and paralogy indicates two genes or proteins are paralogous if they diverged by gene duplication. So, we know two new terms one is the speciation and other is that gene duplication.

So, homologous indicates that they share a common ancestor, then orthologous which indicates that share a common ancestor but how the separation has occurred, they are diverged during the gene speciation and paralogy or paralogous, how the paralogous situation happens because they are diverged by gene duplication.

(Refer Slide Time: 29:05)

Orthology and Paralogy

Homology: Two genes or proteins are homologous if they share a common ancestor.

Orthology: Two genes or proteins are orthologous if they diverged by speciation.

Paralogy: Two genes or proteins are paralogous if they diverged by gene duplication.

Gene duplication (α, β)

Speciation (Human, Mouse)

Globin (α -globin (Human α -globin, Mouse α -globin), β -globin (Human β -globin, Mouse β -globin))

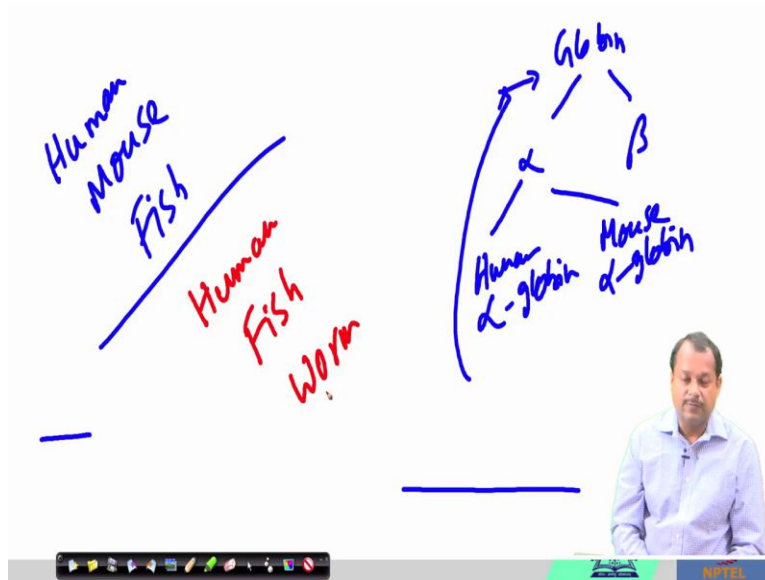
Handwritten notes on the slide: "Globin", "Human α -globin", "Mouse α -globin", and "Human β -globin".

Pralay Mitra

Now, if I make a plot, then I will get one tree structure something like this globin now the gene duplication because of the gene duplication this globin, so this globin because of the gene duplication is divided into two parts alpha globin and beta globin. So, what I can say globin alpha beta. And this part is called as the gene duplication, because of the gene duplication alpha and beta has created.

Now, because of the speciation, so species has been created, so from this alpha human alpha globin and mouse alpha globin has created, for beta also it is similar. So, one is the gene duplication because of the duplication alpha globin and beta globin created and because of the speciation and gene speciation, that human alpha globin and mouse alpha globin has created and the human beta globin and mouse beta globin has created.

(Refer Slide Time: 30:39)



Now, in this context one more relevant point is that during the evolution or considering the phylogenetic tree, then who is closer to say whom. Now if I give you say human, mouse and fish and if I give you say human, then fish, then worm. Now you see that for human, mouse and fish human is distantly related with fish or human, fish, worm, human is closely related with fish.

So, context also matters. So, in which context I am discussing, so that also matters. So, in one case fish is distant and fish is distant with respect to the human, in another case fish is close to the human. So, this is also comes in the context and that is why when one multiple sequence alignment information is given to you and you wish to actually incorporate one more sequence, then the sequence weight will matter.

And that will also depend on the situation like whether you wish to go for more diverged situation or you wish to go for more similar situation, accordingly you give the weight corresponding to that one. So, that is it for now, we will next discuss another such weight value, which will give the weight for the sequences during this alignment that is Henikoff weight and after that one we will move to our structural element, actually which is our intention, and in order to complete that discussion, we move on to the sequence alignment. Thank you very much.