

**Algorithms for Protein Modelling and Engineering**  
**Professor. Pralay Mitra**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Kharagpur**  
**Lecture 59**  
**Summarizing Protein Folding and Protein Docking (Contd.)**

Welcome back. So, we are summarizing the protein folding and protein docking technique. So, we started on the last lecture and we are continuing. On the last lecture we have covered the in silico techniques like protein homology modeling and then we started protein threading. The protein threading we will continue here.

(Refer Slide Time: 00:31)

**CONCEPTS COVERED**

- Homology Modeling
- Protein Threading
- Ab initio Protein Folding
- Tertiary structure to quaternary structure
- Primary structure to quaternary structure
- Current status

Pralay Mitra

So, the concept covered I kept same, but actually we covered homology modeling. We are discussing protein threading.

(Refer Slide Time: 00:41)



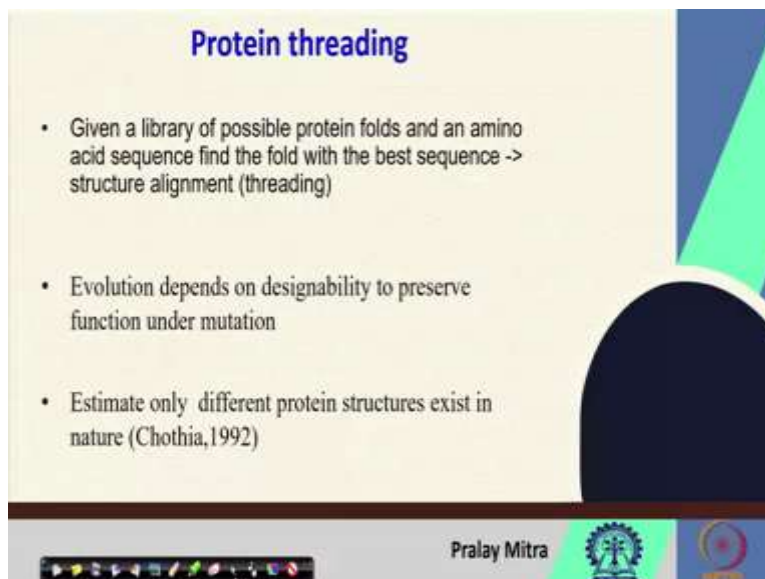
**KEYWORDS**

- Folding
- Docking

Pralay Mitra

The keywords are also kept as same, folding and docking.

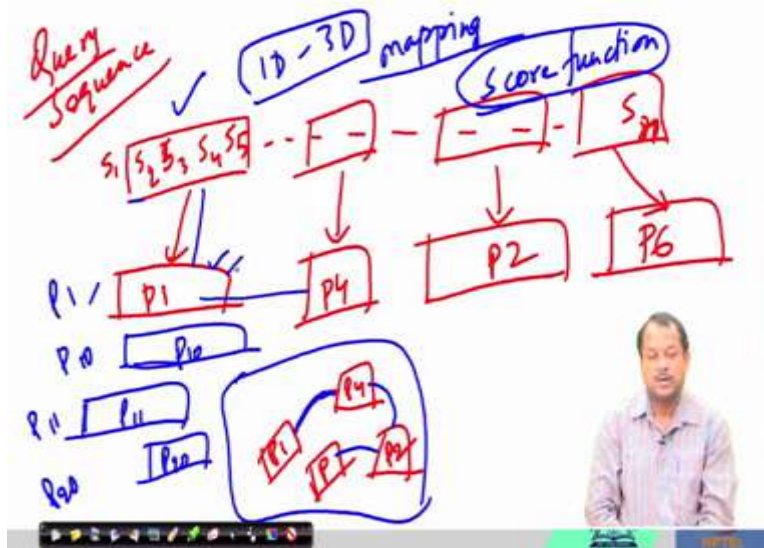
(Refer Slide Time: 00:45)



**Protein threading**

- Given a library of possible protein folds and an amino acid sequence find the fold with the best sequence -> structure alignment (threading)
- Evolution depends on designability to preserve function under mutation
- Estimate only different protein structures exist in nature (Chothia, 1992)

Pralay Mitra



So, formally, if I define then basically given a library of possible protein folds and an amino acid sequence find the fold with the best sequence to structure alignment that is the threading. Now, the evolution depends on designability to preserve function under mutation that is one observation. And estimate all the different protein structures exist in nature. And also, so long back in 1992, Professor Cyrus Chothia has this observation and he mentioned that one.

After that one the study indicates about 1300 folds exist in nature and also the new protein structures which are deposited in the PDB are not novel, I mean, that this is matching with some of the existing one. So, with these observations, actually, people started to look for the protein threading problem. And also if you remember that on the last class, last lecture, actually, what I mentioned regarding this protein threading is say given one query sequence, so you have one query sequence, so when it is a query sequence I can say  $S_1, S_2$ , so  $S_n$  sequences are there.

Now, first job is to identify the fragments. So, when you identify the fragments, then you got something like this. Because what I mentioned given this sequence or query sequence as an input, if you can able to identify another sequence in the protein databank with more than say 70 percent sequence identity, then you are getting homologous sequences. So, directly you can copy most of the structural information for that sequence which is in the PDB and with which your query sequence is having more than 70 percent sequence identity.

But the problem starts when actually it is not the case. So, globally, I mean, that when we go for global sequence alignment we find that sequence similarity is very less. Then we have to go for

or look for some local alignment. And from that point of view, let us assume that this part is matching with this, say this part is matching with this, this part is matching with this and say this part is matching with this and this is taken from say protein 1, this is protein say 4, this is protein 2, this is protein 6 so from different proteins. If you get then your protein threading problem will be definitely fast to identify these structural fragments P1, P4, P2, P6.

Next, you have to organize this say P1, P4, P2, P6 and then you need to connect them in order to get this fold. So, that is in a very short statement what is the protein folding problem. But what are the challenges? There are a lot of challenges. First of all, I, and so this finding this overlapping information or say assembling those information so that is proved as NP complete problem that in detail I will tell you that.

Next thing is that, so with this part say I am getting a match with the protein P1 in small part, but it may possible that along with that one there is another protein say P10 which is matching like this and with another one, say it is matching say P11, with another one say P20 it is matching like this. So, out of P1, P10, P11, P20, which is the best, for that you need to go for 1D to 3D mapping or specifically score function. Now, as good as your score function in order to identify that which fragment should go properly here based upon that one you will get the best match and accordingly you will have the best model structure.

So, that way the design of the score function is also important. Which score, 1D to 3D, because this is 1D and this is 3D structure. You may argue corresponding to the structure there is a sequence. What about aligning with that sequence? In that case only the dynamic programming will work. I agree in one point, yes, dynamic programming will work, but given the structural information also if you do not exploit that one, then probably you are losing some information. So, you have to plan so that you can use sufficient amount of structural information also along with the sequence that is my suggestion.

(Refer Slide Time: 06:32)


**Algorithm 22:** **Protein threading**

**Input:** Given a library of possible protein folds and an amino acid sequence  
**Output:** Find the fold with the best sequence -> structure alignment (threading)

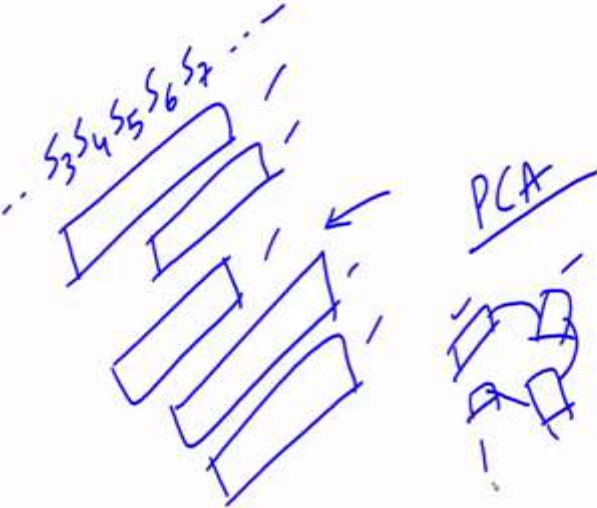

**Steps:**

1. A library of protein folds (templates) ←
2. A scoring function to measure the fitness of a sequence -> structure alignment
3. A search technique for finding the best alignment between a fixed sequence and structure
4. A means of choosing the best fold from among the best scoring alignments of a sequence to all possible folds


PCA



Pralay Mitra



PCA




**Algorithm 22:** **Protein threading**

**Input:** Given a library of possible protein folds and an amino acid sequence  
**Output:** Find the fold with the best sequence -> structure alignment (threading)

**Steps:**

1. A library of protein folds (templates)
2. A scoring function to measure the fitness of a sequence -> structure alignment
3. A search technique for finding the best alignment between a fixed sequence and structure
4. A means of choosing the best fold from among the best scoring alignments of a sequence to all possible folds


**I-TASSER**



Pralay Mitra



CASP



So, here is the algorithm on protein threading. Input, given a library of possible protein folds and an amino acid sequence; output, find the fold with the best sequence to structure alignment or threading that is what is required. So, this is, as you understand, it is not end to end like sequence to structure modeling.

So, for that on the last lecture I mentioned that several steps are they are given one protein sequence then you need to identify the matching secondary, matching fragments, then you have to build one template library, that matching template you can search for PDB from FSSP, from SCOP, from CATH so different data sets you can search for, then you have to identify some score function, then you have to go for threading. So, all those steps are involved.

But only one part I picked that is given a library of possible protein folds and an amino acid sequence find the fold with the best sequence to structure alignment that is threading. So, the steps, a library of protein folds or the template that will be given definitely one innovative idea you have to come up with at this position so that your templates are good enough for your purpose.

It should not be more than required. It should not be less than required. It should be optimum and good enough for your own purpose. Next, a scoring function to measure the fitness of a sequence to structure alignment, that is very important. So, you have to design some scoring function. A lot of scoring functions exist, again so physics based, evolutionary information based so, and combined one exists.

We noted in the context of protein design that it is not the individual one, but the collective score function perhaps doing good, but when we will combine those then we understand that at the individual level we have to check the importance of each of the components and we have to also check that whether they are basically correlated or not, because that may bias something. So, individuals say principal component analysis or any sort of analysis is required in this purpose. So, principal component analysis I say, PCA.

Now, third point is that, a search technique for finding the best alignment between a fixed sequence and structure. So, once you will find that one, then you are almost done. And fourth, a means of choosing the best fold from, among the best scoring alignment of a sequence to all possible fold. That is what I mentioned last time. So, given one sequence say S3 in between, some between sequence I am considering, S3, S4, S5, S6, S7 and I have one fold like this, another like this, another like this, another like this, another like this, so which one to pick.

So, whether I shall go by the coverage or I shall go by the accuracy or which one. So, among these which one to pick? Because when I will go for the threading, if you remember, so I threaded like this, then it was connected here, connected here, connected here, so 1, 2, 3, 4, four fragments are being actually combined in order to get the fold. So, that is what is mentioned here. So, the fourth point, a means of choosing the best fold from among the best scoring alignments of a sequence to all possible folds. So, that is required. So, if you have that then possibly you are done with this.

So, I-TASSER is one of the most popular technique for protein threading based predictions and because of its performance in the CASP also, so in this context I would like to mention that name CASP, so initially it was started like in competition, now it has been a standard kind of a norm. So, once you design some protein folding software, then you should run that one on the CASP data, so which provides a lot of sequence and structure data for benchmarking your method. So, it is the critical assessment of structure prediction. So, initially it was started by John Murt and now it has become a very popular benchmarking competition for protein folding technique.

(Refer Slide Time: 11:28)

**Scoring Schemes for Sequence to Structure Alignments**

- The scoring scheme for a particular threading of a sequence onto a structure measures the degree to which
  - Environmental preferences are satisfied
  - Different amino acid types prefer different environments
  - Structural preferences: helix, sheet, not exposed to solvent
  - Pairwise interactions with neighbouring amino acids

Pralay Mitra

Now, the scoring scheme, so on the last slide when I mentioned you had to design a very good score function, then obviously the question will come in your mind what is the definition of a very good score function. Yes. So, few things you need to consider when you are designing the scoring scheme. So, those few things I am listing here.

The scoring scheme for a particular threading of a sequence onto a structure measures the degree to which environmental preferences are satisfied, different amino acid types prefer different environments, structural preferences like helix, sheet, not exposed to solvent, those things are considered, pairwise interactions with neighboring amino acids will be considered.

Now, you see that most of the components we have discussed in different contexts for protein folding, for protein interaction, for secondary structure prediction, even for phosphorylation site prediction. So, you have to consider the evolutionary situation, you have to consider the



environment, you have to consider their pairwise interaction, you have to consider their secondary structure information, you have to consider their solvent accessibility, all those things you have to consider. And considering all those things only give you a balanced score function.

So, there is a such no guidance for designing a score function, but all the different components like environmental preference, different amino acid type, physicochemical information that includes here, structural preferences like helix, sheet, coil or say buried, intermediate, exposed, then pairwise interaction with neighboring amino acid, what is the sequence level environment, what is the window size. So, those things you have to consider. And corresponding to those things, some component must be there in your score function, perhaps then only you can have a better score function for your purpose.

(Refer Slide Time: 13:29)

**Computational Complexity of Finding an Optimal Alignment**

- The complexity of the protein threading problem depends on whether:
  - (i) Variable-length gaps are allowed in alignments
  - (ii) the scoring function for an alignment incorporates pairwise interactions between amino acids

–Property(I) makes the search space exponential in size to the length of the sequence

–Property(II) forces a solution to take non-local effects into account

Any protein threading scheme with both properties is NP-complete (3-SAT Lathrop 1994) (MAX-CUT Akutsu, Miyano 1999)

Pralay Mitra

So, to make it a complete one, so I wish to bring this to your kind attention also that what is the computational complexity of finding an optimum alignment. So, it is NP complete in nature. So, most of the biological problems are NP complete in nature. So, for some of them there are proofs, there are there are proofs for protein folding, there are proofs for protein design problem. But if it is NP complete or NP hard, then it is intractable as per the computer science, but what to do. So, we cannot sit idle. So, we have to come up with some heuristic.

So, we have to go for some random or some stochastic modeling process through which we will get some solution with reasonable accuracy. And as I mentioned that we are not going for the

solution, rather than we are providing a list of provable solutions from which one is supposed to be the correct one. So, the rest or the final part actually is upon the biologists. But definitely we can encode some rules, some logic which exists in chemistry or biology and based upon that one we can implement and run on say high speed computers so that quickly we can get some suggestions out of that computational techniques. And trust me the accuracy is really very good.

The time complexity of protein threading problem depends on whether the variable length gaps are allowed in alignment or not. So, you may, you know that when it is the alignment problem and mostly the dynamic programming you will be using for this purpose, then there will be a gap, which means you have to go up or go left instead of diagonal. When you are going diagonal then you are aligning in the dynamic programming if you remember. If you go up then there is a gap in the, basically, there is gap in the main columns. When you are going left then there are gaps on the rows.

Second point, the scoring function for an alignment incorporates pairwise interactions between amino acids. Now, the property 1 makes the search space exponential in size to the length of the sequence. And property 2 forces a solution to take non-local effects into account. Regarding to several variations exist and people are still working that whether a better one can be done or not, but improvement is going on.

Needless to mention that this particular problem is also proved as NP complete reducing from 3-SAT problem and MAX-CUT problem both and the two papers. Their citations are mentioned here. If you are interested you can go through that paper.

(Refer Slide Time: 16:35)

**Comparison with homology modeling**

- Both are TBM, no boundary in terms of prediction techniques.
- But the protein structures of their targets are different.
  - Homology modeling (HM) is for those targets which have homologous proteins with known structure (usually/maybe of same family).
  - Protein threading (PT) is for those targets with only fold-level homology.
  - Thus, HM is for "easier" targets and PT is for "harder" targets.
- Homology modeling treats the template in an alignment as a sequence, and only sequence homology is used for prediction.
- Protein threading treats the template in an alignment as a structure, and both sequence and structure information extracted from the alignment are used for prediction.

Pralay Mitra

Now, we have discussed today in the previous lecture and this lecture two in silico protein folding techniques, one is the homology modeling, another is that protein threading or fold recognition. Now, in both cases, there is role of homologous sequences as well as the structure. In case of homology modeling, globally or at the whole level I am getting the sequence similarity and in case of protein threading small, small fragments I am getting as similar then I need to thread them. So, template library is required for both of them.

Now, how it differs? Both are TBM. So, both are based upon template based modeling. So, templates are required for homology modeling as well as for protein threading. Now, no boundary in terms of prediction techniques. I mean, that I mentioned that if it is greater than 70 percent, then you should go for homology modeling, less than 70 percent protein threading as such one hard threshold is difficult to mention, but if you get adequate number of homologous sequences whose structure is also known, I mean, homologous sequence from the protein databank that actually implicitly translates that you have the homologous sequences whose structures are known.

So, if you have adequate number of such cases and they are globally also homologous with each other, then probably you have to go for, you can go for homology modeling. Otherwise, you may have to go for this protein threading or fold recognition problem, but the protein structure of their targets are different. So, homology modeling HM is for those targets which have homologous

proteins with known structure usually or maybe if same family as for the SCOP class. Protein threading PT is for those targets with only fold-level homology. So, at the fold-level they are same.

So, thus, HM is for easier targets and PT is for harder target. Then again the word within double quote “easier” and the “hard”, so these two terms are also considering the fact that what will be the accuracy of your protein modeling. So, if the accuracy is very high, which means it is probably the easier target. If not then probably it is the harder target. Now, homology modeling treats the template in an alignment as a sequence and only sequence homology is used for prediction. Protein threading treats the template in an alignment as a structure and both sequence and structure information extracted from the alignment are used for the prediction.

(Refer Slide Time: 19:28)



One new artificial intelligence based technique has come. So, it is the AlphaFold. So, developed by Google's DeepMind, AlphaFold predicts the protein structure utilizing artificial intelligence based technique. And perhaps this artificial intelligence will take over that homology modeling and protein threading also the ab initio where a lot of computations and lot of things we have to consider, so probably, no, we have to focus on some artificial intelligence based program and then accuracy will be very high. And as of now the accuracy of the AlphaFold is very high compared to other existing techniques. So, let us see how far we can go with this AlphaFold.

(Refer Slide Time: 20:16)

**Protein-protein docking**

FTDock  
ZDOCK/ZRANK  
PatchDock/FireDock  
ClusPro

Input: Two protein structures  
Output: Protein complex

*FFT*  
*Further sorting and re-ranking Tertiary*  
*functional*  
*Quaternary*

**Protein-protein docking**

FTDock  
ZDOCK/ZRANK  
PatchDock/FireDock  
ClusPro

Input: Two protein structures  
Output: Protein complex

*SymmDock*  
*Fiber Dock*  
*Geometric Hashing*

Pralay Mitra

So, with that we summarize the protein folding problem that we have discussed. As I mentioned, a lot of problems or a lot of programs and the algorithm behind that we did not able to finish, but we picked a few of them and we discuss that one. Now, in case of protein-protein docking, so if we summarize that what we have discussed, so first of all the definition of the protein indicates that two protein structures are given as an input you have to output a protein complex.

So, to do that one we identified, if you remember, if I give you the summary of that one, that given two protein structures and you need to identify that what will be their complex. Now, there can be any complex, but here I should add the word functional complex. So, it is not the case that

if I combine these two and get one structure, then that is my protein complex and that is the output of protein docking, no.

So, I have to predict one structure that structure must be functional in nature or it should have some function. So, that is why here protein structure is the input which means tertiary structure and quaternary structure is my output. Now, in this context, we started with some base technique in order to generate the different orientations, then we see that if I apply the fast Fourier transform algorithm on that one then it will have a huge speed up on that one. So, that way I can generate a lot of decoys.

After generating those decoys, we perform grid based, we perform say surface complementary based, interface packing based, geometric packing at the protein interface or the protein contact area. Interface I will say when it is functional form. So, at the contact area I identify that one. And after identifying that one, then I score them, and then I sort based upon the score value in order to get the rank.

And I mentioned, instead of one solution you provide set of 5 or top 10 based upon the score function. So, the technique or the software which exists on that one, one is the FTDock which relies on fast Fourier transform FFT based algorithm, then ZDOCK and ZRANK both coming from Zhiping Weng's lab.

So, this ZDOCK also uses the similar concept of FFT and ZRANK is on top of this ZDOCK, once you got a list of solutions, then on that list of solutions you can perform some sophisticated scoring for further sorting and re-ranking. So, this ZRANK will allow you further sorting and re-ranking. So, these two are mostly, in the complex generation point of view, these two are mostly similar. This PatchDock is completely different paradigm. So, it relies on this PatchDock geometric hashing.

We discussed in detail also this geometric hashing which was initially taken from the concept of the protein, computer vision and used in this protein docking. Now, this PatchDock was the basic algorithm which actually generates that different decoys and then it scores are different decoys based upon their geometric fitting or in the actually the geometric hashing also there is a provision that you can give as a fingerprint that what are the features based upon that one it computes.

Nevertheless, similar to this ZDOCK on top, ZRANK on top of ZDOC, so FireDock is also on top of PatchDock. So, PatchDock provides a list of solutions. On those list of solutions, FireDock do side chain refinement. Now, if you remember our example of bound and unbound docking, so unbound docking is more realistic. What it used to do that it considers two separately crystallized protein molecules and identifies that complex.

When they are separately crystallized so their side chains are also optimized accordingly. Unlike two bound complexes where I am taking out two chains and then giving some random orientation, but their side chains are optimized in complex form, whereas for the unbound form the side chains are optimized separately. So, even if you find the correct orientation, you may find that at the geometry the fitting is not correct, probably the score function is not that much good.

So, the FireDock allows some sort of flexibility at the side chain once you get one provable solution. It is sort of a very fast kind of side chain fitting or side chain optimization algorithm you can consider. We also mentioned two other variations. So, one is changing the backbone of that one that is FiberDock, although I am not sure about the performance of this FiberDock, and symmetry based degeneration of the docking that is SymmDock, so which we did not discuss or talk about this.

And there exists another protein-protein docking software that is called as a ClusPro that mostly relies on clustering the decoys generated by some technique say either geometric hashing or say based upon brute-force technique followed by the fast Fourier transformation. So, based upon that one the decoys you have generated you cluster and based upon that clustering analysis, it outputs the solution.

So, the accuracy of the ClusPro is also very good since it relies mostly on that clustering technique. Now, this is actually the protein-protein docking starting from the protein structure, so which means tertiary structure is given to you as an input you are predicting the quaternary structure.

(Refer Slide Time: 27:16)

**Protein-protein docking**

*Primary Structure*

Input: Two protein sequences

Output: Protein complex

*Q. Structure*

Two approaches:

1. Model the input protein sequences and then use the modeled (folded) protein structure for docking using any of the existing protein docking techniques.
2. Sequence -> Protein complex (Directly)

Pralay Mitra

But if you are given the protein sequences, protein sequences means, you are given protein sequences, which means primary structure, and then if you are predicting quaternary structure, then how can you do that one. If you can do that one then perhaps you can identify the protein complex for a number of proteins, because the number of protein sequences in nature is very high compared to the number of protein structures. The structure, determining the structure is a very lengthy and complicated process either x-ray crystallography or NMR or say (( ))(28:11) using that one you can determine the structure. So, that is the lane the process.

That is why in the PTB which houses experimentally, experimental structures is very less compared to total number of protein sequences which are available in nature. So, if you can have one computational lot ready which will take two protein sequences as an input and then predict what will be the structure, I mean, the complex structure that will be of enormous use. One simplest way you can think, given two protein sequences, I can basically model them separately two different protein folding, then I will go for that docking. That is fine. That is one solution.

Another, directly from the sequence is it possible for you to predict the protein complex structure. So, yes, it may possible if corresponding to those two sequences you will find homologous protein structures in the protein bank, protein databank and that homologous structures are also forming a complex in the protein databank. So, that might be one solution.



(Refer Slide Time: 29:30)



So, these are the some of the things we have discussed regarding the protein folding and protein docking. Lot of things are also there. So, you can explore them. And so that is it for this lecture. Thank you very much.