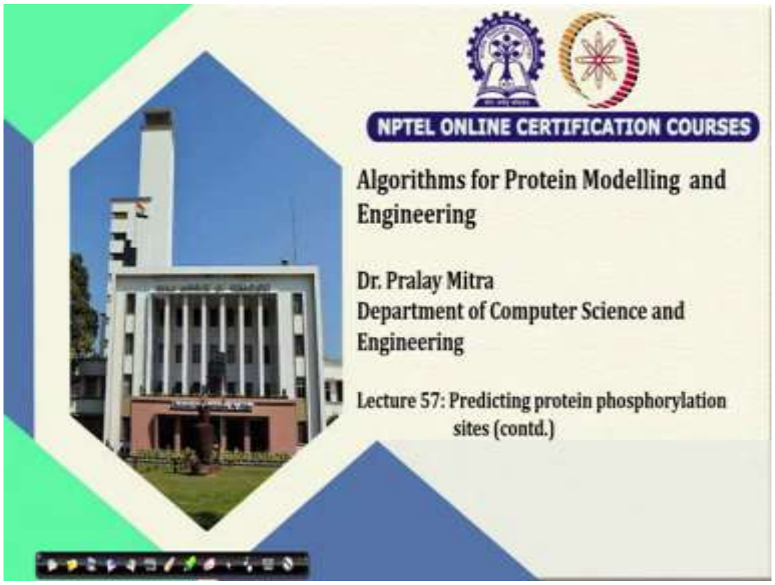
Algorithms for Protein Modelling and Engineering
Professor Pralay Mitra
Department of Computer Science and Engineering,
Indian Institute of Technology Kharagpur
Lecture 57
Predicting Protein Phosphorylation Sites (Contd.)

Welcome back and welcome to the class of Algorithms for Protein Modeling and Engineering.

(Refer Slide Time: 00:21)



NPTEL ONLINE CERTIFICATION COURSES

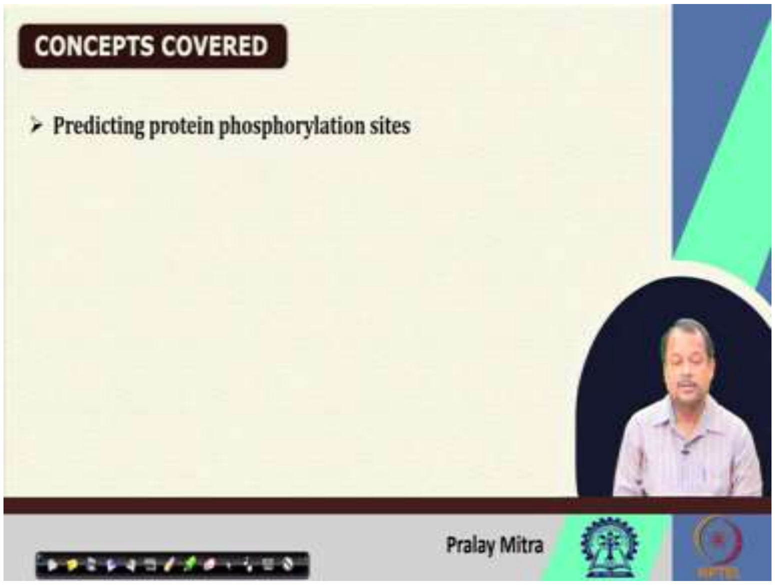
Algorithms for Protein Modelling and Engineering

Dr. Pralay Mitra
Department of Computer Science and Engineering

Lecture 57: Predicting protein phosphorylation sites (contd.)

So, we are continuing our discussion of predicting phosphorylation site for proteins or predicting protein phosphorylation sites.

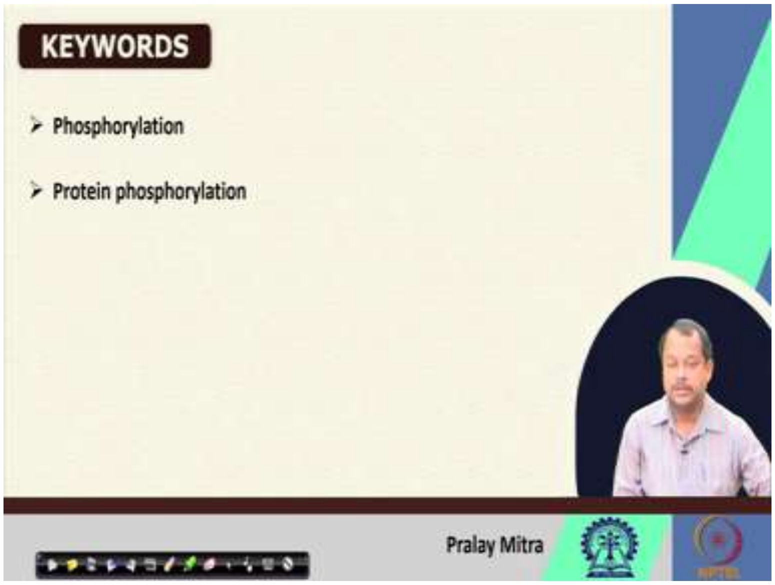
(Refer Slide Time: 00:29)



CONCEPTS COVERED

- Predicting protein phosphorylation sites

Pralay Mitra



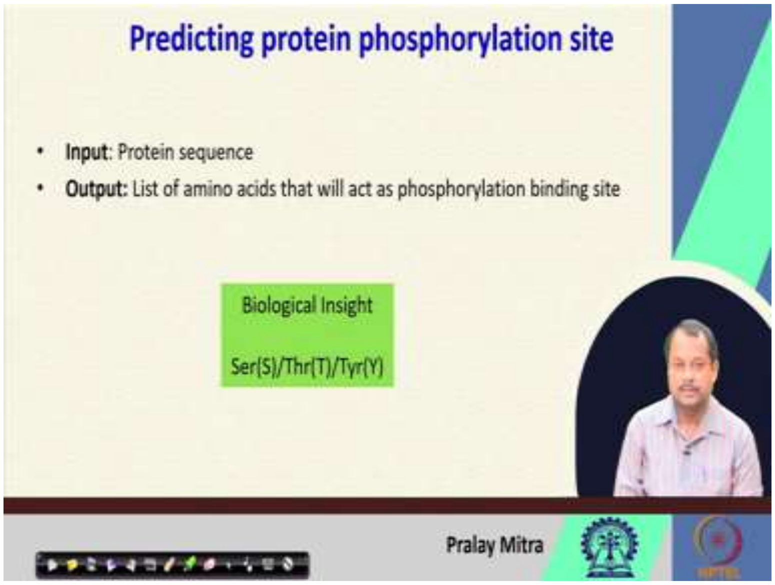
KEYWORDS

- Phosphorylation
- Protein phosphorylation

Pralay Mitra

So, we are continuing our discussion from the last lecture where we described a little bit of chemistry and the biology and then we came to the conclusion that specifically it is the serine, threonine and tyrosine, S, T and Y who are capable of performing the phosphorylation process. So, if we are interested to predict the phosphorylation site from a protein then we have to focus only on these three amino acids. So, for the rest of the amino acids, it is not required. That way we are pruning down the search space heavily and also improving the prediction accuracy and that way also I am explaining to you that some sort of domain knowledge is always required for this kind of algorithm development.

(Refer Slide Time: 01:20)



Predicting protein phosphorylation site

- **Input**: Protein sequence
- **Output:** List of amino acids that will act as phosphorylation binding site

Biological Insight

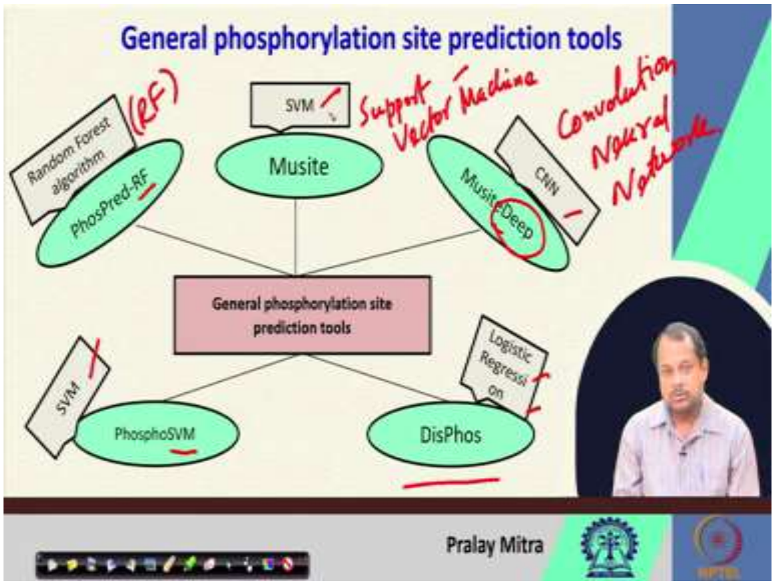
Ser(S)/Thr(T)/Tyr(Y)

Pralay Mitra

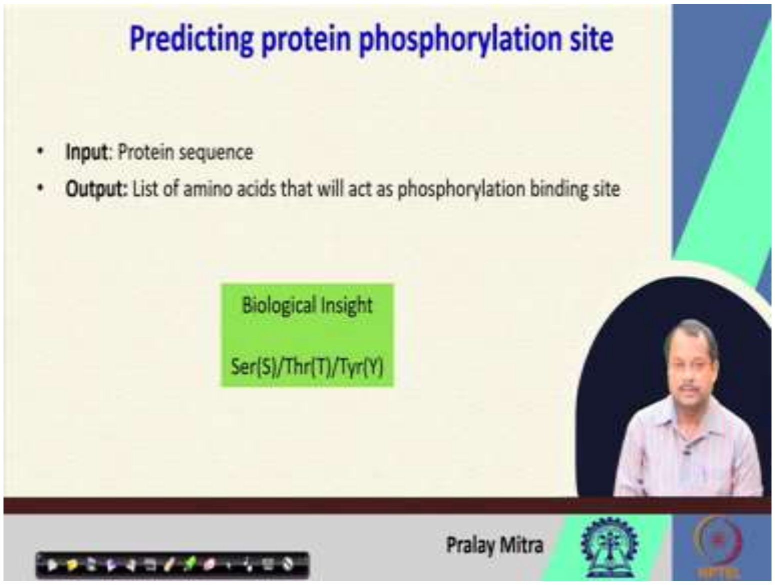
So, if I think of predicting protein phosphorylation site, so the input is protein sequence and the output is list of amino acids that will act as phosphorylation binding site. And the biological insight that we derived from the last lecture is it is going to be the serine, threonine or tyrosine S or T or Y who are capable of forming the phosphorylation, who are capable of acting as the phosphorylation site.

So, I am not telling that all the S, T and Y will be the phosphorylation site. I am telling that it is the subset of the S, T and Y only who will act as the phosphorylation site. So, other amino acids or residues do not have any role in this prediction. So, any role means that in the prediction as a feature set they will definitely come, but they will not act as a phosphorylation site that is what is my proposal based upon the domain knowledge.

(Refer Slide Time: 02:27)



General phosphorylation site prediction tools



Predicting protein phosphorylation site

- **Input:** Protein sequence
- **Output:** List of amino acids that will act as phosphorylation binding site

Biological Insight

Ser(S)/Thr(T)/Tyr(Y)

Pralay Mitra

Now, we also discussed that there are a lot of software tools which has developed for predicting the phosphorylation site. Grossly they are categorized into two parts; one is the kinase specific, another is the general phosphorylation site prediction tools. So, the kinase specific we are not discussing that one, because it is based upon what kind of kinase it is. So, some sort of domain knowledge again is used for that. But we are interested about rather more challenging problem where that kinase information is not present, then what to do.

And as I mentioned in this slide that input is a protein sequence. So, from this point of view that it is bit challenging also in the sense that, when the structure is given to you as an input, then along with the biological insight that S, T and Y is going to be the phosphorylation on site, then you can use another biological insight that it must be on the surface, because phosphorylation site is the functional activity related PTM and since it relates with some function, then it must be on the surface.

Now, if the structure is known to you then using some NACCESS or Connolly surface that we have discussed or even using the DSSP that we discussed on the last week, you can able to identify whether that particular amino acid is on the surface or not. So, combining that information along with this biological insight like S, T and Y is going to be the phosphorylation site, it perhaps is easy to develop some model or technique or tool. But if we start with a protein sequence, then also, then it is little bit challenging.

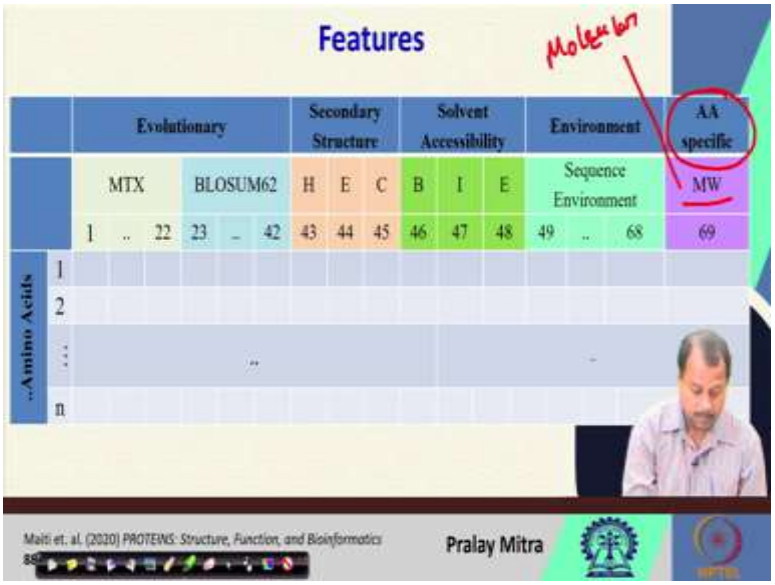
But at the same time considering the fact that the number of protein sequences are high or very high compared to the number of protein structures which are available then from an application point of view, if you start with a protein sequence, I mean, if you develop some algorithm or tool where the input is a protein sequence, then it is going to be of much use compared to if you start with the protein structure.

So, almost all the techniques are based upon this protein sequence and do not use any other structural information. Here I am telling you that most of the techniques are relying on the machine learning or deep learning techniques. So, let us start with the Musite, which is developed using the SVM or support vector machine, here Musite.

This PhosphoPred-RF as the suffix in the name also suggests that is based upon the random forest algorithm, this possible SVM again as the suffix of the name suggests, based upon this SVM or support vector machine, this phos is based upon the logistic regression model, MusiteDeep an advanced version of this Musite actually uses the deep learning technique. So, the suffix indicates that deep and that indicates deep learning technique. So, they have used CNN or convolution neural network.

Now, we will see some other machine learning tool, so which can actually compete with the CNN or SVM or other RF or logistic regressor techniques. So, it is true that sometimes CNN or deep learning can do miracle in terms of the accuracy and can do great. But here we will see that if we can compete using some features and if we then probably we can able to identify some features which will be useful for the biologist to go inside the problem and analyze that what is the reason behind this phosphorylation. So, we will try to do that one.

(Refer Slide Time: 07:18)

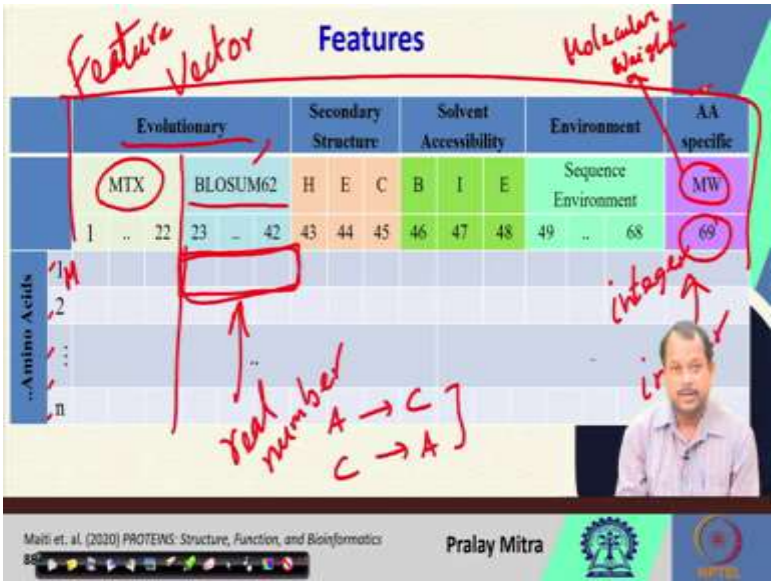


Features

		Evolutionary						Secondary Structure			Solvent Accessibility			Environment		AA specific
		MTX			BLOSUM62			H	E	C	B	I	E	Sequence Environment		MW
		1	..	22	23	..	42	43	44	45	46	47	48	49	..	68	69
..Amino Acids	1															
	2															
	⋮					..									-		
	n															

Maiti et. al. (2020) PROTEINS: Structure, Function, and Bioinformatics

Pralay Mitra

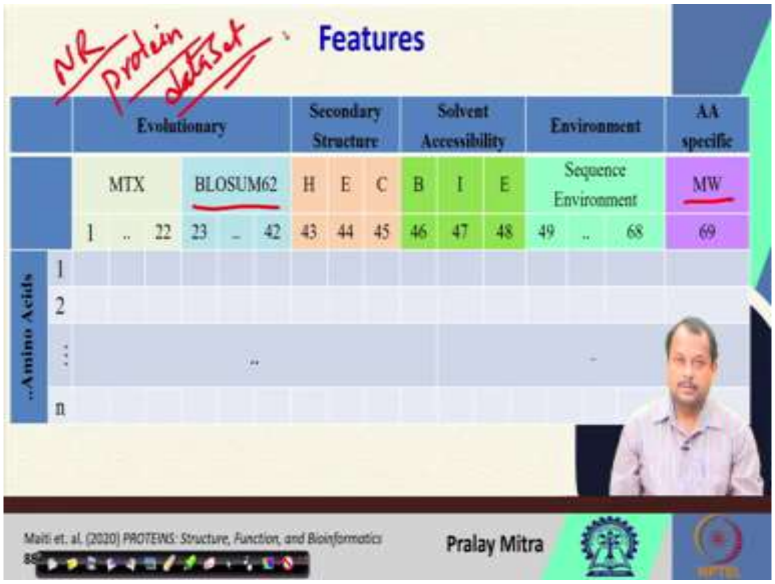


Features

		Evolutionary						Secondary Structure			Solvent Accessibility			Environment		AA specific
		MTX			BLOSUM62			H	E	C	B	I	E	Sequence Environment		MW
		1	..	22	23	..	42	43	44	45	46	47	48	49	..	68	69
..Amino Acids	1															
	2															
	⋮					..									-		
	n															

Maiti et. al. (2020) PROTEINS: Structure, Function, and Bioinformatics

Pralay Mitra



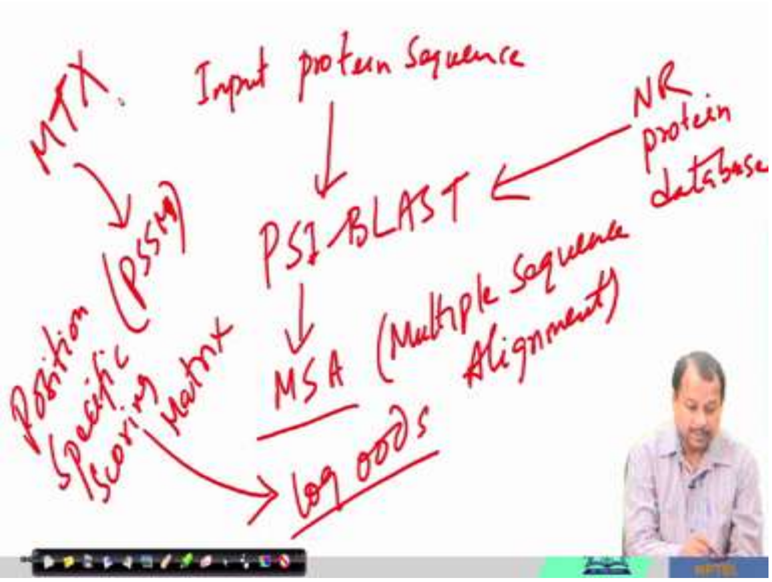
NR protein dataset

Features

		Evolutionary						Secondary Structure			Solvent Accessibility			Environment			AA specific
		MTX			BLOSUM62			H	E	C	B	I	E	Sequence Environment			MW
		1	..	22	23	..	42	43	44	45	46	47	48	49	..	68	69
..Amino Acids	1																
	2																
	:					..									-		
	n																

Maiti et. al. (2020) PROTEINS: Structure, Function, and Bioinformatics 88

Pralay Mitra



MTX. Input protein Sequence

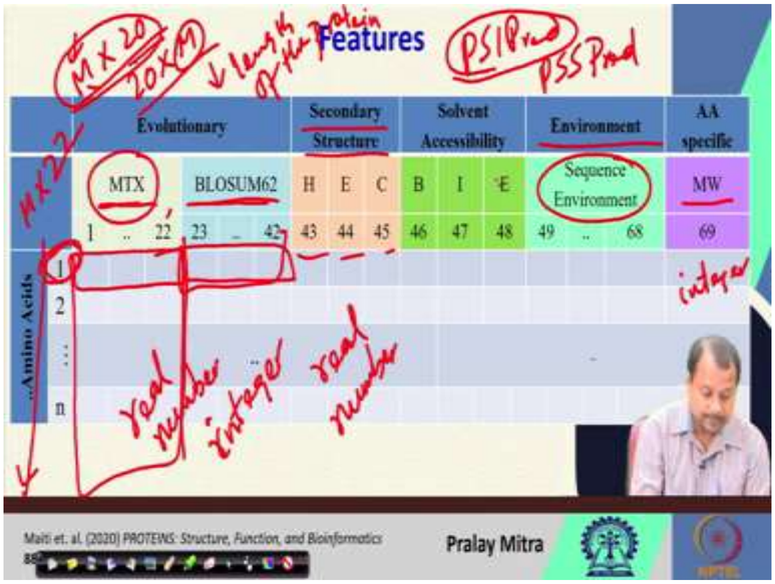
↓

PSI-BLAST ← NR protein database

↓

MSA (Multiple Sequence Alignment)

MTX → Position Specific Scoring Matrix (PSSM) → log odds

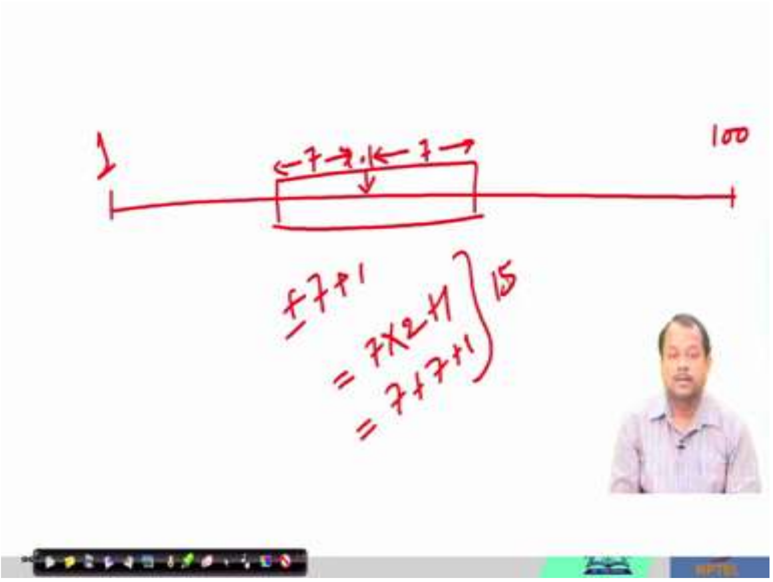


Features

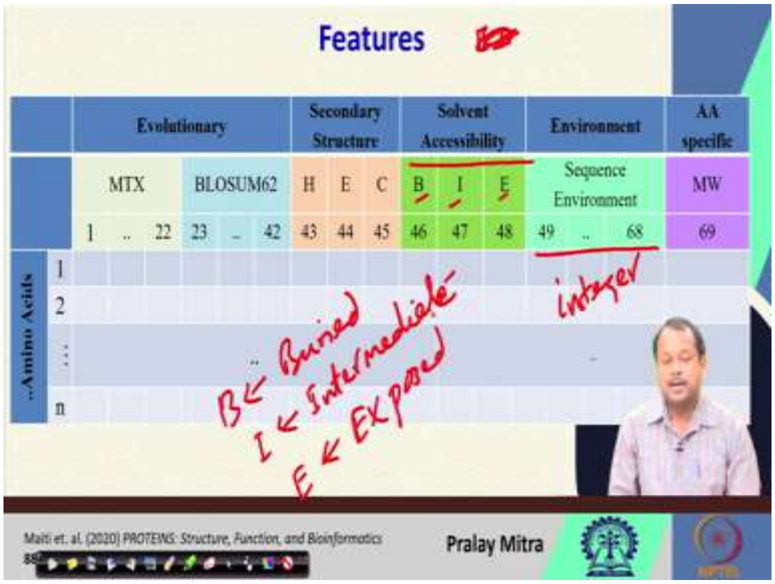
		Evolutionary						Secondary Structure			Solvent Accessibility			Environment			AA specific
		MTX			BLOSUM62			H	E	C	B	I	E	Sequence Environment			MW
		1	..	22	23	..	42	43	44	45	46	47	48	49	..	68	69
..Amino Acids	1																
	2																
	:					..									..		
	n																

Maiti et. al. (2020) PROTEINS: Structure, Function, and Bioinformatics 88

Pralay Mitra



$$\left.\begin{aligned}&\pm 7+1\\&=7\times 2+1\\&=7+7+1\end{aligned}\right\}15$$



Features

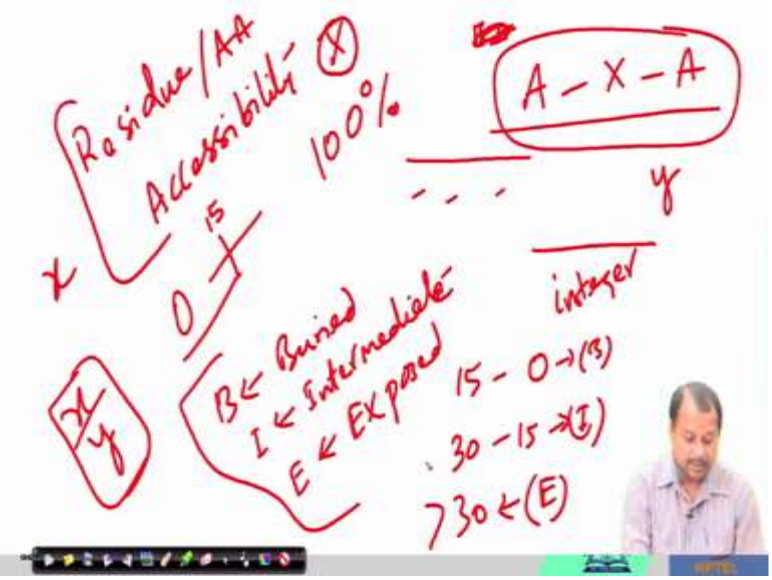
		Evolutionary						Secondary Structure			Solvent Accessibility			Environment			AA specific
		MTX			BLOSUM62			H	E	C	B	I	E	Sequence Environment			MW
		1	..	22	23	..	42	43	44	45	46	47	48	49	..	68	69
..Amino Acids	1																
	2																
	⋮						..								..		
	n																

B ← Buried
I ← Intermediate
E ← Exposed

integer

Maiti et. al. (2020) PROTEINS: Structure, Function, and Bioinformatics

Pralay Mitra



Residue/AA Accessibility (X)

100%

A – X – A

y

x

15

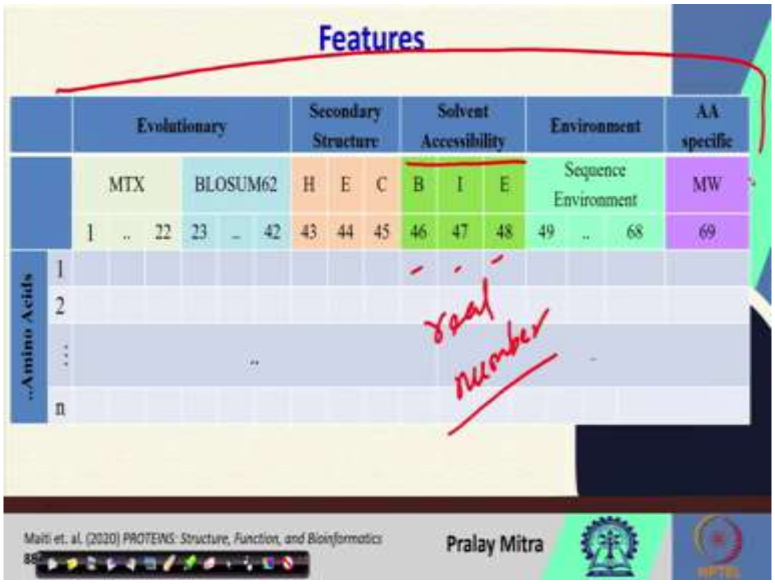
0

$\frac{x}{y}$

B ← Buried
I ← Intermediate
E ← Exposed

integer

15 - 0 → (B)
30 - 15 → (I)
>30 ← (E)



Features

		Evolutionary						Secondary Structure			Solvent Accessibility			Environment			AA specific
		MTX			BLOSUM62			H	E	C	B	I	E	Sequence Environment			MW
		1	..	22	23	..	42	43	44	45	46	47	48	49	..	68	69
..Amino Acids	1																
	2																
	:					..								-			
	n																

real number

Maiti et. al. (2020) PROTEINS: Structure, Function, and Bioinformatics 88

Pralay Mitra

So, since I mentioned that we are going to say design one machine learning based tool, so we need the feature first. So, what are the features? So, the features are grossly classified into five categories. I should say four plus one, because the plus one is basically amino acid specific, the last one, the amino acid specific, and that is the simple thing. It actually lists the molecular weight. MW indicates, this indicates molecular weight. Now, the first one is the evolutionary information.

So, here, two terms are combined one is I am calling as MTX, another BLOSUM62 matrix. So, for the BLOSUM62 matrix, I do not think I need to explain it again. So, we discussed it in the last lectures. So, this BLOSUM62 actually having some evolutionary related information and it says that what is the probability of the substitution by one amino acid to another amino acid. So, that is a 20 cross 20 amino acids we are having.

Now, if I consider all the, so if I consider that mutation of one amino acid to another is actually symmetric in nature, so if I muted say A to C and C to A in both the cases the probability is going to be the same then the BLOSUM62 matrix will not be 20 cross 20 it will be the lower triangular matrix plus the diagonal values.

Now, the diagonal value indicates that what is the probability of mutating somebody by itself. So, if I do not consider that diagonal, then actually it will be the lower triangular matrix because of this mutation. Now, what I will do that as a feature the BLOSUM62 matrix also I will consider. How, I am explaining that.

So, first, input is my protein sequence. Now, when input is my protein sequences then each amino, corresponding to each amino acid, I am creating one feature vector. So, this is my feature vector, where I just explained to you what is the last feature, that is the molecular weight. That is my 69th feature. Now, at the first I am having evolutionary information one is the MTX and other is BLOSUM62.

So, what is BLOSUM62? When I say one amino acid at this position, so these are indicating the amino acid positions, now if I assume the first amino acid is methionine then in the BLOSUM62 matrix corresponding to methionine what are the probabilities of other amino acids mutated by the methionine that will be copied and pasted here. So, 23, 24, 25, 26, 27 that way all amino acids probability will be placed here. So, 20 such probabilities will be placed here, including methionine.

So, that is going to be your feature at this position. So, this feature is going to be your real number. So, these BLOSUM62 matrix will be your real number. And these amino acid specific information like the molecular weight that is going to be your integer. The molecular weight generally we consider as integer. So, that is going to be your integer. So, these two things I explained to you, one is the BLOSUM62 matrix, another is the molecular weight.

Now, about the rest, so these two I covered, this one and this one. Now, next regarding the MTX file. So, in the MTX file what I am going to do, you remember that during the PSIPRED on the last week, when we are predicting the secondary structure from the protein sequence, then one intermediate file was created and I mentioned that how I got that one.

So, the first step of the PSIPRED or that neural network method was that, given one protein sequence as an input your job will be to run PSI-BLAST three iterations and you get the homologous sequences from say protein databank and after getting that one now for the secondary structure I was running that one for the, against the protein databank.

But you may think that, okay, I am running the PSI-BLAST against the non-redundant data set, so NR protein dataset and in this case, specifically, we do not have any bound that we should know what will be the secondary structure, we should know what will be the structural information, that is why I should not restrict myself to the protein databank which contains less number of data compared to the non-redundant protein dataset which is the dataset of the sequences only.

Now, this NR protein dataset against that one I can perform the PSI-BLAST. Now, after doing the PSI-BLAST so the log odds of the multiple sequence alignments will be created. To make it more clear to you, so step by step what will be done is, so first what I will do that input sequence then PSI-BLAST with NR protein database. With NR protein database, I am doing the PSI-BLAST. After doing the PSI-BLAST, I am getting MSA or multiple sequence alignment.

Now, from this multiple sequence alignment, I will compute the log odds which is nothing but my position specific scoring matrix, in short PSSM. Now, this PSSM is actually stored as an intermediate file. I am calling that as a MTX file. Actually, in the BLAST this MTX file occurs, is MTX file is there so that MTX file is occurred as an intermediate file. So, here this MTX file I am talking about.

So, this MTX file actually giving the evolutionary profile information corresponding to the given input sequence this BLOSUM62 matrix actually giving the, BLOSUM62 matrix is giving the substitution probability with respect to the evolutionary information for that particular amino acid.

Now, please note it down that this MTX file if you remember correctly will give me M cross 20 or say 20 cross M, so just different. So, that indicates for, so where this M is the length of the protein. Now, this M cross 20 or 20 cross M is being used here. So, 1 through 22 actually what I am doing, actually if you do the BLAST then you will see that two more extra column will come, that is why I kept two more, and what you will see that corresponding to each amino acid so this M, M is the length of the protein sequence, so that will go here and corresponding to each amino acid so 20 I am telling actually it will be 22 as for the BLAST, so that 22 list I am providing here corresponding to each amino acid and that way if you take only this part that is going to be MTX file of the size M cross 22 actually BLAST gives M cross 22, not 20, M cross 22.

So, directly if you get the input sequence, do the PSI-BLAST, from there you will get that log odds value as the position specific scoring matrix that value you copy directly so that matrix you copy directly and place it. So, based upon in which format you are getting, whether you have to take the transpose or not, I am not sure, you please take that one, and you put it here that MTX directly followed by this BLOSUM. For this BLOSUM actually one standard BLOSUM62 matrix is there with you. So, what you need to do that you copy each row corresponding to each amino acid which amino acid is being considered right now and you place that here. So, that way I finished the discussion on MTX, BLOSUM62 and molecular weight.

After that one what I am insisting that you put the secondary structure. So, extensively on the last week we discussed how to predict the secondary structure from the given input sequence. So, you can use either PSIPRED or PSSpred tool for this purpose of which the algorithm behind PSIPRED is discussed, the algorithm PSSpred is also similar, it differs only in the number of the hidden layers and the nodes in the hidden layer. And both are predicting three different states. And when they are predicting the three different state actually a probability value will come.

So, here my proposal is that instead of telling that what sort of secondary structure that amino acid will represent, you provide the probability value for forming the helix seat and coil, so that is why three different fields are there. So, that way this is going to be your real number. This is going to be your integer, because that BLOSUM62 as you noted that scale to some integer. So, here since it is a probability value, so it is also going to be a real number. And here I mentioned this is integer.

Now, next this environment, this environment is actually sequence environment. Sequence environment indicates that when say one protein sequence is given to you and I am interested to analyze this one for any purpose and say I am considering this particular amino acid at this moment because along the rows each amino acids are present there.

So, if I am considering this amino acid then I will consider one sequence environment like this. So, you remember for PSIPRED we have used say plus minus 7. So, which means from here 7, from here 7 and this itself plus 1, which means 7 cross 2 plus 1 or 7 plus 7 plus 1, 15. So, 15 amino acids will be computed. So, that environmental information, the way we compute it, will come here.

So, here basically, sorry, here basically which amino acids are present, so that information I will put. Here you can see that there is a space for 20 amino acids. So, 20 amino acids means that, so I will count that within my window if I consider plus minus 7 then considering myself as 1 then within that 15 how many say alanine residues are occurring, how many cysteines are occurring, how many aspartic acids are occurring, how many glutamic acids are occurring so that number or count will come directly here.

That way surely you will get 0 in some cases because in some window there may not be some amino acids. And also if I consider 15 is my window and since 20 different amino acids are there, then few amino acids are definitely going to be 0 that is fine, but it will tell the environment. So, that way it is also integer for me.

Finally, the solvent accessibility, now regarding the solvent accessibility, although we did not discuss, I mean, solvent accessibility we discussed in the context of protein structure. So, when protein structure is given to you then how to compute the solvent accessibility using say Connolly's algorithm or say NACCESS program or say DSSP that we mentioned. But if a protein sequence is given to you, what will be the solvent accessibility that we did not discuss.

Before going to that one let me tell you, so B, I, E, what is that. So, B indicates buried, let me write it here, I intermediate and E exposed. So, here what is done that one thing you can consider that first I will compute in one say, so for example, let us consider structure. So, with respect to structure I am going to define this B, I, E, then actually what I will do that I will combine that information B, I, E for the sequence also. So, for this B, I, E so for a structure you are actually computing the accessible surface area.

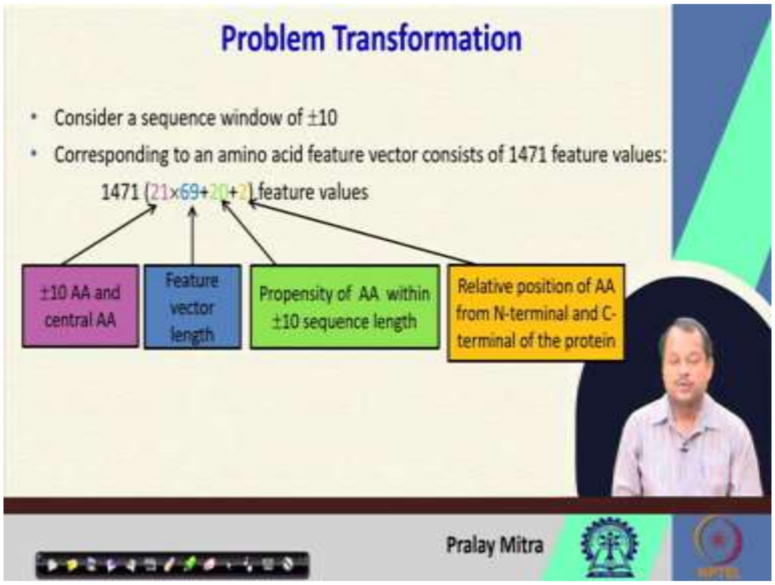
Now, I mentioned that you compute based upon the atom. And so, atom-wise accessibility is known to you. Then what he will do, you take the summation over the atom in order to get residue level or amino acid level accessibility. When you say that or when you get the residue level accessibility then you got some value let us assume that is my x. Now, if without any loss of (())(24:11) I assume that name of the residue is x. I know that x cannot be one residue. But you know that for any algebraic equation when it is unknown we generally consider that as x. From that point of view I am assuming x. x is the residue.

And for capital X residue small x is going to be the accessibility I got in that particular protein structure. Then one model I will consider that if I consider that there is one tripeptide A-X-A. So, one alanine, another say my residue, I am considering, then another alanine and there is a peptide bond between the A to X and X to A. So, if I have that tripeptide then in this tripeptide bond what is the accessibility of this x, let us assume that is y, then I will do x by y. So, this fraction will tell me that what percentage of the residue is basically exposed.

If I assume 0 to 100 is the accessibility, so when it will be 100, when it will be same as y. So, when it will be same as y, y divided by 1, so percentage will be 100. Then I can put some criteria like say 0 to 15, so 15 to 0 that is going to be your B. Then I can consider say 30 to 15 is going to be my I and greater than 30 is going to be my E, which means that if the accessibility is less than 15 then I will say buried, when it is more than 30 I will say exposed and in between or intermediate is in between. Based on that I have three definitions B, I, E.

Now, similarly, the way we have actually predicted the secondary structure using the similar technique what we can do we can have one such neural network technique which can predict the solvent accessibility and that probability value directly I can put here. So, that way it is also going to be my real number. And that concludes the discussion over this whole feature vector.

(Refer Slide Time: 26:51)



Problem Transformation

- Consider a sequence window of ±10
- Corresponding to an amino acid feature vector consists of 1471 feature values:

1471 (21×69+20+2) feature values

±10 AA and central AA

Feature vector length

Propensity of AA within ±10 sequence length

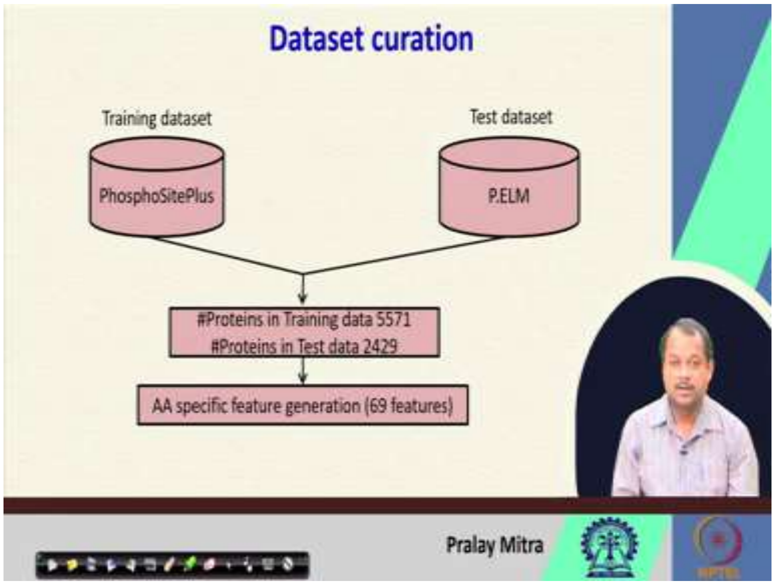
Relative position of AA from N-terminal and C-terminal of the protein

Pralay Mitra

Now, this feature vector we are computing for all the amino acids in a protein sequence and the window length we consider plus minus 10, where PSIPRED has considered a plus minus 7, so we consider plus minus 10, then it will be 20 number of amino acids, 21 including me. Now, corresponding to an amino acid then the feature vector will be consisting of 1471 feature values. What are those?

So, first 21 based upon that sequence information, sequence window, then 69 is the feature that we computed just now, then 20 propensity of amino acid within plus minus sequence length, then relative position of amino acid from N terminus and C terminus of the protein. So, that relative position information also we find important and that is going to be another feature. In total 1471 feature values corresponding to one amino acid is being computed.

(Refer Slide Time: 27:58)



Dataset curation

Training dataset

PhosphoSitePlus

Test dataset

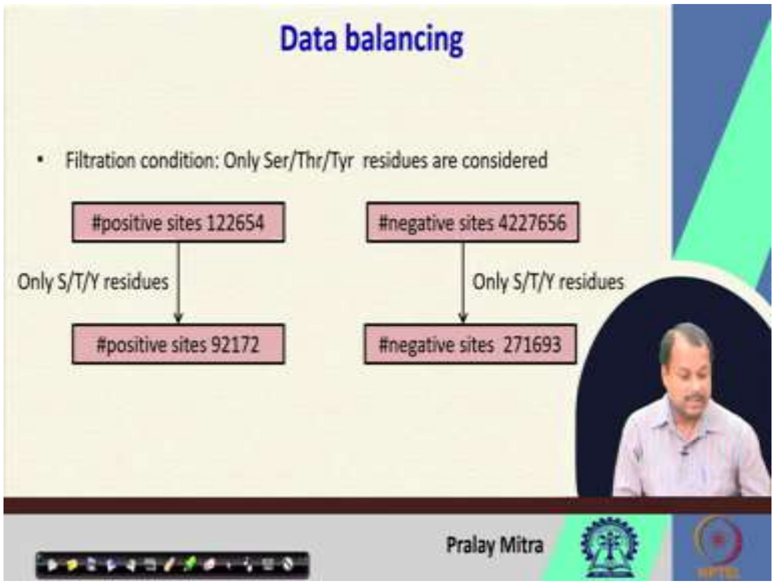
P.ELM

#Proteins in Training data 5571
#Proteins in Test data 2429

AA specific feature generation (69 features)

Next, the dataset is curated where PhosphoSitePlus and P or phosphate dot ELM, phospho dot ELM those two datasets are there. So, P dot ELM is a more authentic data. That is why we consider that as the test dataset and training data set is PhosphoSitePlus. Now, combining those we have proteins in the training data 5571 and proteins in test data is 2429. Now, amino acid specific feature generation. So, we generate 69 features. For that, detail discussion we have done.

(Refer Slide Time: 28:36)



Data balancing

- Filtration condition: Only Ser/Thr/Tyr residues are considered

#positive sites 122654

Only S/T/Y residues

#positive sites 92172

#negative sites 4227656

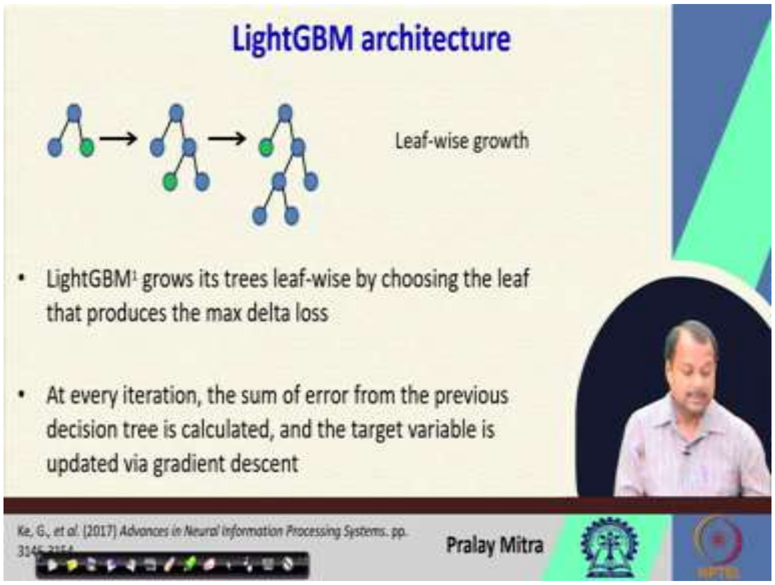
Only S/T/Y residues

#negative sites 271693

Pralay Mitra

And then we consider some data balancing also. Using the biological insight we consider only serine, threonine or tyrosine residues, where the positive sides are 122,654 and negative sides so 4,227,756. Now, if I consider only serine, threonine and tyrosine then it will be further reduced to 92,000 positive and say 271,000 negative, still there is not balancing in the data. So, we need to go for balancing.

(Refer Slide Time: 29:12)



LightGBM architecture

Leaf-wise growth

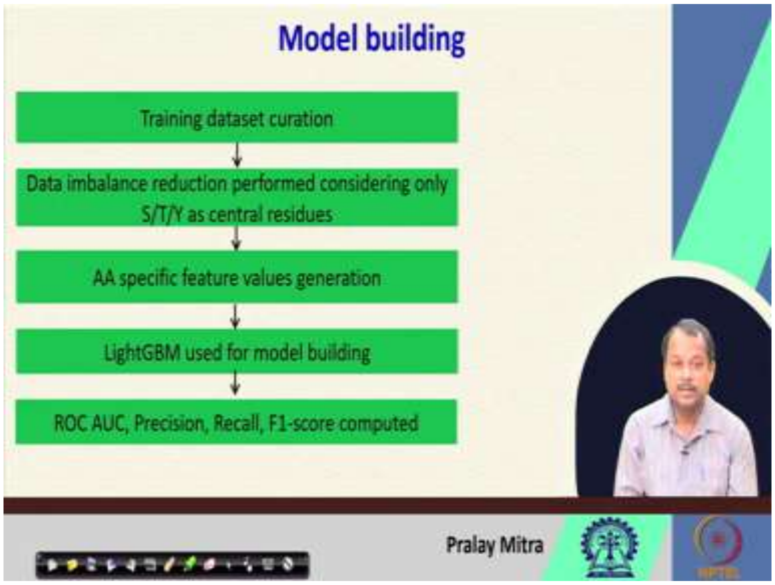
- LightGBM[1] grows its trees leaf-wise by choosing the leaf that produces the max delta loss
- At every iteration, the sum of error from the previous decision tree is calculated, and the target variable is updated via gradient descent

Ke, G., et al. (2017) Advances in Neural Information Processing Systems. pp. 3146-3154

Pralay Mitra

And then we use the LightGBM architecture. So, if you consider the LightGBM architecture then what is the (())(29:26) I will show you, but here why you can use the LightGBM is because it grows leaf-wise and choosing the leaf that produces max delta loss. At every iteration, the sum up error, at every iteration, the sum of the error from previous decision tree is calculated and the target variable is updated via gradient descent technique.

(Refer Slide Time: 29:47)



Model building

Training dataset curation

Data imbalance reduction performed considering only S/T/Y as central residues

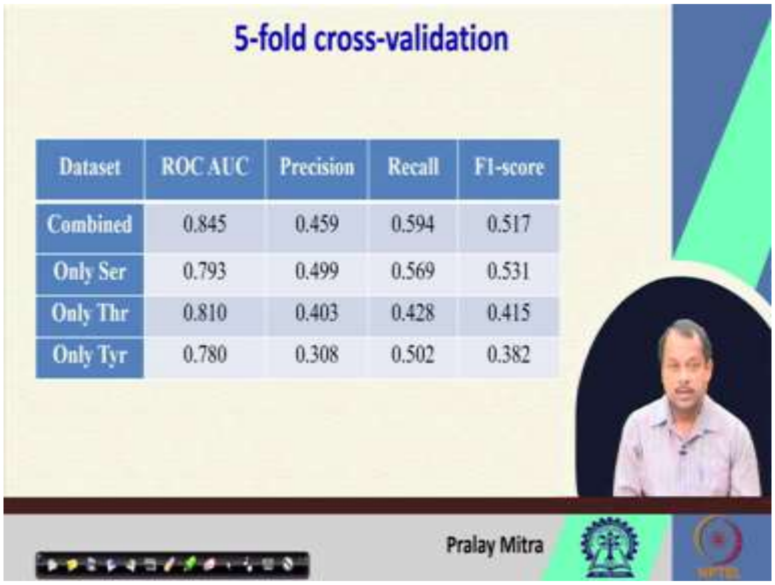
AA specific feature values generation

LightGBM used for model building

ROC AUC, Precision, Recall, F1-score computed

So, the model building is done this way training data set curation, data imbalance reduction performed concerning only S, T, Y as central residues, amino acid specific feature values are generated, LightGBM used for model building and four models has been created one for serine, another for threonine, another for tyrosine, another for combined and we will see that what is the effect on that one.

(Refer Slide Time: 30:09)



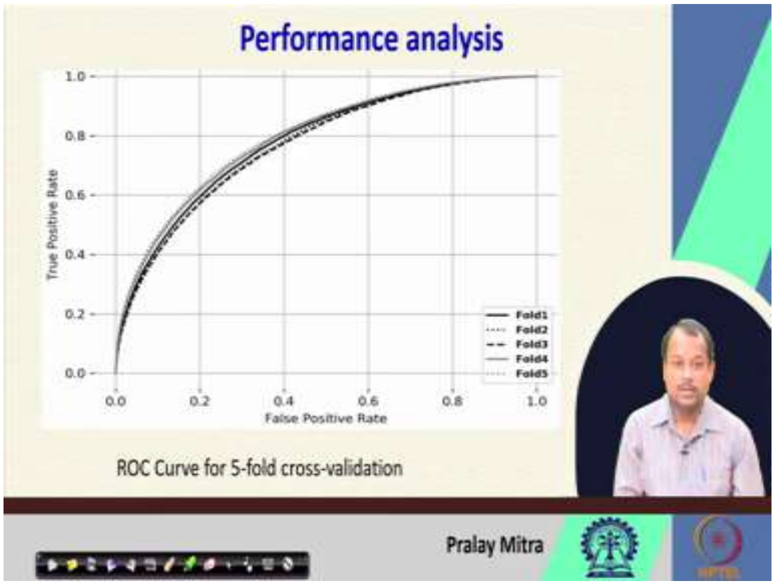
5-fold cross-validation

Dataset	ROC AUC	Precision	Recall	F1-score
Combined	0.845	0.459	0.594	0.517
Only Ser	0.793	0.499	0.569	0.531
Only Thr	0.810	0.403	0.428	0.415
Only Tyr	0.780	0.308	0.502	0.382

Pralay Mitra

Here you can see. When we see that only serine the performance is not that much good, but only threonine performance as per the ROC AUC is very good. But if I look at the F1-score then serine is better. So, it is 0.53 and actually you should look F1-score for better accurate F1-score and that way serine is the best, tyrosine is the worst and combining everything if I look at, then F1-score is going to be 0.517 that is the fivefold cross validation. But this time you know what is the fivefold cross validation. So, I am not going into details of that one.

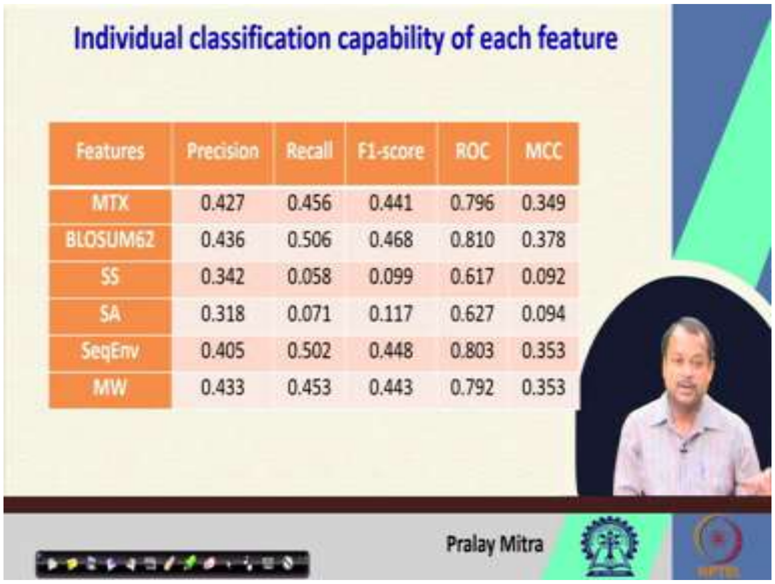
(Refer Slide Time: 30:51)



Performance analysis

ROC Curve for 5-fold cross-validation

Pralay Mitra



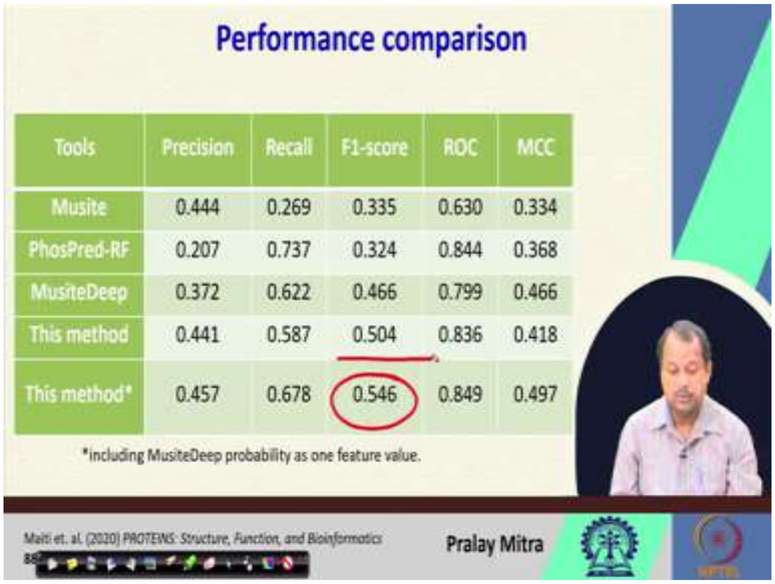
Individual classification capability of each feature

Features	Precision	Recall	F1-score	ROC	MCC
MTX	0.427	0.456	0.441	0.796	0.349
BLOSUM62	0.436	0.506	0.468	0.810	0.378
SS	0.342	0.058	0.099	0.617	0.092
SA	0.318	0.071	0.117	0.627	0.094
SeqEnv	0.405	0.502	0.448	0.803	0.353
MW	0.433	0.453	0.443	0.792	0.353

Pralay Mitra

Nevertheless, you have to look for the, this is the ROC curve of the plot for that particular algorithm and individual classification capability of each feature also you need to analyze. So, either like this way say precision, recall, F1-score, ROC, MCC or using say some box plot you can do that analysis. At the gross level the MTX, BLOSUM62, secondary structure, solvent accessibility, sequence environment and molecular weight has been considered. So, each value is not considered, the features are only considered.

(Refer Slide Time: 31:23)



Performance comparison

Tools	Precision	Recall	F1-score	ROC	MCC
Musite	0.444	0.269	0.335	0.630	0.334
PhosPred-RF	0.207	0.737	0.324	0.844	0.368
MusiteDeep	0.372	0.622	0.466	0.799	0.466
This method	0.441	0.587	0.504	0.836	0.418
This method*	0.457	0.678	0.546	0.849	0.497

*including MusiteDeep probability as one feature value.

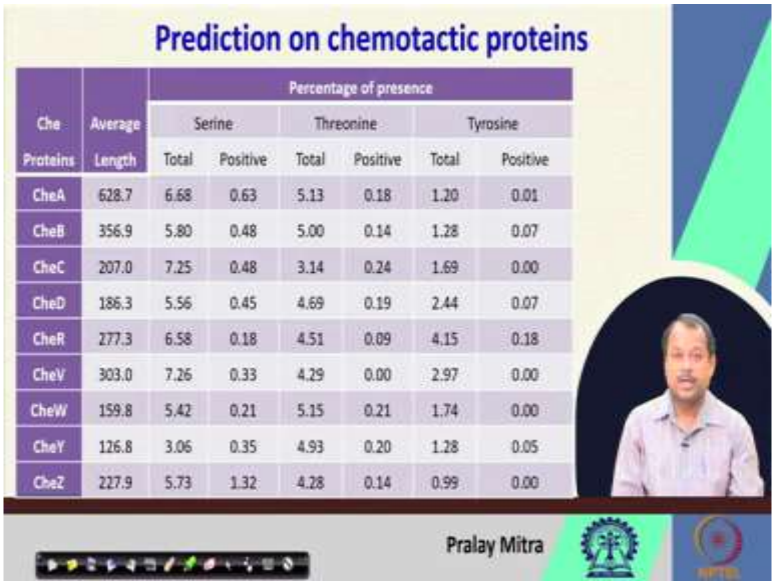
Maiti et. al. (2020) PROTEINS: Structure, Function, and Bioinformatics

Pralay Mitra

So, here is the performance comparison of this method along with the Musite, PhosphoPred and MusiteDeep. So, there are two variations of this method. As you see that there is an option for adding the feature values, so the output of some another technique can be added as one feature technique. So, if I add that one then the accuracy will increase. So, that is that this method star, where the F1-score is 0.546 that is the highest one. So, sorry, this is the highest one.

But if I do not include that one, only the feature values that we consider then also it is 0.504. So, compared to other techniques this is not bad. So, this we have achieved based upon only the machine learning. But definitely there is a scope of improvement and you can work with that one.

(Refer Slide Time: 32:17)



Prediction on chemotactic proteins

Che	Average	Percentage of presence					
		Serine		Threonine		Tyrosine	
Proteins	Length	Total	Positive	Total	Positive	Total	Positive
CheA	628.7	6.68	0.63	5.13	0.18	1.20	0.01
CheB	356.9	5.80	0.48	5.00	0.14	1.28	0.07
CheC	207.0	7.25	0.48	3.14	0.24	1.69	0.00
CheD	186.3	5.56	0.45	4.69	0.19	2.44	0.07
CheR	277.3	6.58	0.18	4.51	0.09	4.15	0.18
CheV	303.0	7.26	0.33	4.29	0.00	2.97	0.00
CheW	159.8	5.42	0.21	5.15	0.21	1.74	0.00
CheY	126.8	3.06	0.35	4.93	0.20	1.28	0.05
CheZ	227.9	5.73	1.32	4.28	0.14	0.99	0.00

Pralay Mitra

So, this is the prediction on chemotactic proteins that I mentioned. So, some examples are given here.

(Refer Slide Time: 32:27)

So, that is it for this discussion. Thank you for your attention.