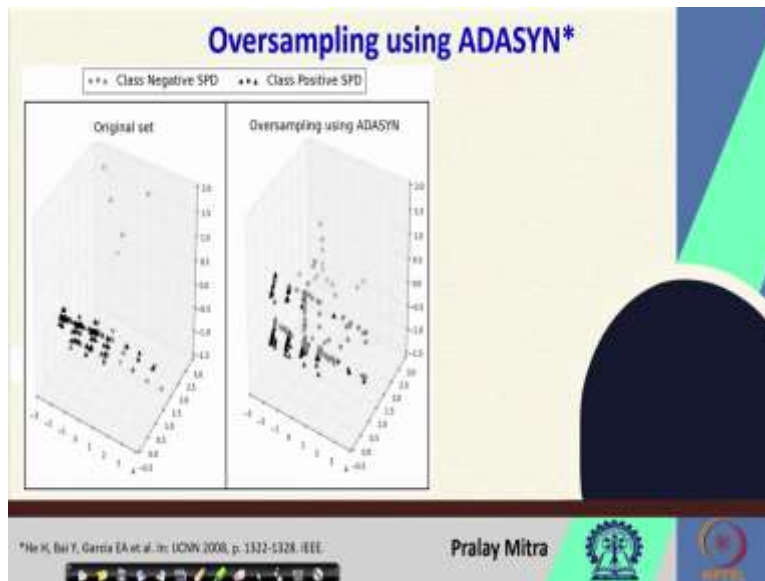


Algorithms for Protein Modelling and Engineering
Professor Doctor Pralay Mitra
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur
Lecture 49
Protein Modification (Contd.)

Welcome back. So, we are continuing with the protein modification, specifically protein engineering. So, we started the discussion on single point deletion. And for the single point deletion the dataset, the feature set that we discussed. Now we are planning to go for the classifier design.

So, we will be covering the single point deletion, and after that will be moving to multi point deletion. Both part of protein modification or protein engineering. So, we are also mentioning that as the protein modification.

(Refer Slide Time: 00:59)



So, this is the InDel operation. So, one problem was there. So, we have 132 positive features for which actually, for which protein L plus protein S was present, and 30 negative cases for which protein L was present from the literature. But if I corresponding to one particular, say amino acid, if we delete then it will unfold. That way we are having 162 total number of data.

Now this 132 plus 130, sorry, 132 plus 30, so that is not a balanced data. And it is known to all of us that if the dataset is not balanced then there is a possibility that the result will be biased also. That is why what we are now trying to do? That we are creating some new negative samples,

because negative data are in short for us. So, we are creating some negative data by doing oversampling using the ADASYN. So, ADASYN has developed long back and it has lot of use, specifically for oversampling using ADASYN.

In this plot you can see class negative single point deletion and class positive single point deletions are there. So, from the original set when the class negative cases are less and class positive are more, then in this case when you see that class negative cases has increased and almost there is a balance between this class negative and class positive.

(Refer Slide Time: 02:58)

Classifying foldability on SPD instances

Classification Performance (LOOCV)

P1 → P32
N1 → N30

Training Data Set	Classification Method	True		False		Accuracy (%)	Misclassified SPD Instances
		Positive (%)	Negative (%)	Positive (%)	Negative (%)		
Class balanced using ADASYN	Random Forest	99.2	0	100	0.8	99.4	P114
Positive instances from SPD_DB	Elliptic Envelope	97.7	0	100	2.3	98.2	P73, P82, P126

Banerjee et. al. (2019) Journal of Proteome Research 18(3):1402-1410 Pralay Mitra

Next, we that when we are all set, we have the feature. We have the dataset. The dataset is also balanced. Partly it has taken; the dataset was a problem because partly it was taking from the literature. But you know that people do not report the unfolded structure in the literature. And also when it is an unfolded structure what does that mean? Is this a problem with the experiment? Or what will it convey? That is why I getting negative data for protein engineering or protein modification is a big challenge.

Fortunately we got 30 and from that 30, we extended few more using oversampling technique by ADASYN. And from the PDB, Protein Data Bank we curated and we got 132 pairs. So, with that we are classifying the foldability on protein instances.

So, classification performance we compared with the LOOCV, leave one out cross validation, and then in the training dataset we did two different kinds of training and testing, or cross validation rather. One is with the class balanced using ADASYN, another positive instances from SPD_DB. And let us see that how it will differ.

So, the classification method is one is the Random Forest for class balanced with the ADASYN. When there was no problem because it contains equal number of say, or proportional number of positive and negative case. So, Random Forest, or in short, RF method is being used. Then positive percentage or true positive case 99.2, false positive is 0. True negative is 100 and false negative is 0.8. And accuracy is 99.4 and misclassified SPD instance is only one, this is P114.

So, basically that 162 that we got and after adding with the ADASYN, so those are marked. And if you mark then there is, you can give some serial number basically. So, all the positives start with P, negatives start with N. And then there will be a serial number. Since 132 positive cases are there. So, P1 will start from P1 to P132, or negative N1 through N30. But fortunately there is no negative instance which has been misclassified.

Now, for positive instances from SPD_DB, when we consider only the positive instances because if you feel that ADASYN is a, sometimes that if you also feel or the other people think ADASYN is generating some proteins, I agree, but doing some data analysis, but what is the physical or biological relevance with biology. So, that sequence may not exist in nature. Again, so when it is unfolding which means that it does not exist in nature but that is say, hypothetical or say, virtual or not existing sequence.

So, if that is the case then you start with only the positive instances from SPD_DB then you have only P1 through P132. And then it is not the classification, rather outlier detection problem. So, how many outlier cases are there? And we are using elliptic environment, elliptic envelope. So, in elliptic envelope what we got that 97.9, 97.7 percent true positive cases are there. Again 0 false positive, 100 percent true negative; 2.3 false negative is there. Accuracy is 98.3. Here are also you see that three cases are there out of 132 which got misclassified by these SPD instances. So, that is in summary of the effort on going for single point deletions.

(Refer Slide Time: 07:24)

Protein stability subject to InDel operation

Stable?

Deletion of residues 33 to 35

Will the protein attain its native fold?

Stable?

SPD(33)
SPD(34)
SPD(35)

33, 34, 35, 36, 37

Multi-point deletion (MPD).

Is this same as SPD?

Pralay Mitra

Now, it is time to move on to multi-point deletion. As I am mentioned, single point deletion is not same as the multi-point deletion. So, multi-point deletion needs separate attention with respect to the single point deletion. So, from this problem wise you can see that it is mostly same. So, deleting residues x to y, so this entire part I am deleting. Then basically I am connecting these two. So, these two are connected here. Then the question is this is stable, will it be stable? That is the question.

Again, this we need to perform without doing protein folding, because in protein folding you know that time requirement is very high whereas this machine learning does, that we demonstrated for single point deletion also, you understand that random forest technique or say, elliptic envelope technique will be very fast. And for that you need not have to go for, say high-end computing resources also. In your laptop or desktop also you can do because as a now, we discussed only the machine learning technique, no deep learning also and also no protein folding, etc.

So, is this same as SPD? So, no, it is not, from problem's complexity point of you. So, we need to delete separately. You should not think that multi-point deletion is kind of iterative single point deletion. So, if say, I wish to, for example delete say, 33, 34, 35, 36, 37. Then it is not the case that we will go for SPD of 33, next SPD of 34, next SPD of 35, like that. It is not that. So, the problem is completely different. Please note it down.

(Refer Slide Time: 09:48)

Constructing the MPD dataset (for PU learning)

PDB
↓
Psi-BLAST

12AB (A, B)
homomer (dimer)
A, C, D
homo trimer

Constructing the MPD dataset (for PU learning)

PDB
↓
Psi-BLAST

12AB-A || 1XYZ-P
12AB-B || 1XYZ-P

12AB (A, B)
homomer (dimer)
1XYZ
homo trimer

Pralay Mitra

So, we need to construct the MPD database also for PU learning, positive unlabeled learning. That is the method we are going to discuss. The database creation is the same as for the previous one for single point creation what we have done. So, we have to start with the Protein Data Bank. Then starting from the Protein Data Bank you perform the BLAST. It is the sequence level BLAST. So, basically you have to extract the chains of all the protein structures in the Protein Data Bank. Then you have to do the BLAST between the chains.

But you can use psi-BLAST also which is more powerful compared to the BLAST. After doing that one you retain only those alignments where say, E value is greater than say 10 to the power

minus 5 and coverage is also greater than 50 percent. After retaining that one then you check that whether those structures have some structural irregularities, I mean that missing coordinates or their resolution is not good etc. So, in those cases you please remove those cases.

After that one then you filter for, if the identical cases are there, which means two sequence alignments are there but they are same. It may be possible. So, if I assume that 12 AS, it contains 2 chains, AB and it is a homomer. Now if I assume there is one hypothetical protein, say XYZ, 1 XYZ if I assume this is my hypothetical protein and it contains, say A, C, D and this is homomer. This is dimer. This is homo trimer. Then this A and this is A, this B and this A, the pair wise alignment of say...

Or in order to avoid the confusion, if I say P, Q, R then 12 AS_A when aligned with 1 XYZ_P, that is equivalent to 12 AS_B when aligned with 1 XYZ_P, because both are homomers. So, they are same. In that case I will retain one and remove another. So, that way I need to eliminate that duplicate also. After eliminating that duplicate whatever will be retained is your positive cases. With that I will start working.

(Refer Slide Time: 12:57)

MPD site features- Deletion site specific


- End to end distance between residue X and Y
- Walk through distance between residue X and Y
- Intra-chain Hydrogen bond, Salt bridges and Disulphide bonds
- Max and Min of Phi-Psi angles
- Mean and standard deviation of Accessible Surface Area
- Mean and standard deviation of Loop propensity (for MPD in loops)
- Amino acid propensity for 20 Amino Acids

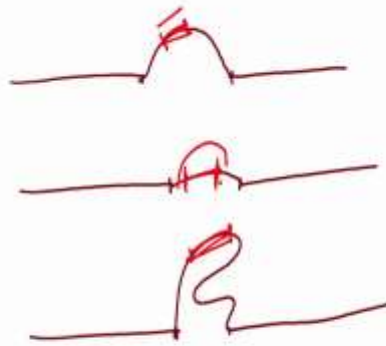
EED
WTD


$$\text{propensity}_x = \frac{\text{count}_{\text{Del_stretch}}^x / \text{count}_{\text{Del_stretch}}}{\text{count}_{\text{Entire}}^x / \text{count}_{\text{Entire}}}$$

W

Pralay Mitra





Now, for this again I have to come up with some feature set. So, in this case MPD site features deletion site specific features. So, you have to identify the deletion site. Now end to end distance between residue X and Y is going to be one feature interesting, because you have to remember that as the distance increases then stitching them together after the deletion will also be a problem. So, if they are very close with each other then it is probably easy. But when they are not then that is going to be a problem.

Now, walkthrough distance between X, Residue X and Y. Walkthrough distance means that when, say, let us consider two situations. One is that this is, another, another. Now in all the cases these are my cut points. Now one thing I need to compute is the distance between two X points, this directly. That if you consider then you will understand that what is the difference between those two.

Next, walkthrough distance between X and Y indicates you have to go this way and what is the distance, you have to go this way and what is the distance, if I have to go this way and what is the distance. Now if I assume this is my EE, end to end distance EED, and this is my walkthrough distance WTD then you will understand that if my walkthrough distance here is something like this, I will delete this one, then the number of residue which will be deleted is large compared to this one and this one. Now if I consider that these are my sides and here is my deletion.

So, I will go to a fresh page. Now if I consider these 3 tests. Now let us assume that red color indicates contiguous multi-point deletion. Now you see that if this is my deletion point, now for this case it may be not that much difficult that after the deletion I will connect these two points. For this case, after this deletion it will be comparatively easier again to connect these two points; whereas for this, after the deletion connecting this with this you have to make a lot of effort. So, these two distances, point to point distance and the walkthrough distance will actually tell how easy will it be from geometry point of view to connect after the multi-point deletion of the contiguous residues.

(Refer Slide Time: 16:38)

MPD site features- Deletion site specific

- End to end distance between residue X and Y
- Walk through distance between residue X and Y
- Intra-chain Hydrogen bond, Salt bridges and Disulphide bonds
- Max and Min of Phi-Psi angles
- Mean and standard deviation of Accessible Surface Area
- Mean and standard deviation of Loop propensity (for MPD in loops)
- Amino acid propensity for 20 Amino Acids

Handwritten notes: $\frac{21}{22}$, $\frac{3}{5}$, $\frac{10AA}{3ala}$, $\frac{3}{10}$, $\frac{5}{75}$

Diagram labels: $\frac{0}{1}$, $\frac{3}{5}$, $\frac{10AA}{3ala}$, $\frac{3}{10}$, $\frac{5}{75}$

propensity = $\frac{\text{count_Exit}}{\text{count_Entry}}$

Pralay Mitra

Next, intra-chain hydrogen bond, salt bridges and disulfide bonds that you can calculate. So, for the Salt Bridge, so there is, you basically compute that Q1 Q2, that Columbic interaction, that same thing you need to compute. For the disulfide bond also you can calculate that what will be the distance. Based upon that one you can do that one.

For hydrogen bond there is a specific criteria. So, for the hydrogen bond what you need? That hydrogen molecule, the donor, acceptor and acceptor antecedent. So, the hydrogen bond will form between the hydrogen and the oxygen or nitrogen on the other side. Now one will be the donor. Another will be the acceptor. So, who is donating? Who is accepting? And based upon that geometry that hydrogen bond strength will depend. So, it mostly is whether hydrogen bond is forming or not, it is a kind of a binary information which is being passed.

But if you wish to make it more sophisticated then you have to compute that angle. And based upon that one, because that hydrogen bond is the non-covalent bond and it has lot of effect on the direction. So, if the direction changes then the affinity or the interaction energy will change for the hydrogen bond. As such the hydrogen bond, individual hydrogen bond is of very low interaction. Again if the angle changes it will also change.

So, although most of the application will count the number of hydrogen bonds, but for the sophisticated calculation it is always preferable to compute the percentage of the contribution based upon their angle. So, definitely I will also insist to count the number of hydrogen bonds, but along with that one, when you are counting you should not add 1 for one hydrogen bond but add 1 multiplied with what fraction it is contributing based upon its orientation or arrangement. Then you will get a sophisticated calculation.

So, max and min of phi angles. So, what is the range of the phi angles that you consider? Mean and standard deviation of accessible surface area, so that will tell you that whether the deletion is on the surface or the core; because on the surface it is sometimes easy to accommodate and it will be mostly stable but if it is going to be at the core then it is difficult to accommodate and the stability will lose definitely. Mean and standard deviation of loop propensity for multi-point deletion in loop. What is propensity? I will come to that one, but before that one amino acid propensity for 20 amino acids.

So, propensity, the definition suggests that in a particular region, so if I consider the amino acid propensity here for 20 amino acids; so in a particular deleted region, so how many times one particular amino acid occurs divided by what is the deletion stretch, whole divided by, in the total protein how many times that particular amino acid x occurs divided by total length of the protein. That is the propensity. That propensity you need to compute for here.

For example, if you are deleting a stretch of, say 10 amino acids and in that stretch 3 alanine occurs. So, the numerator is going to be 3 divided by 10. Now the denominator is going to be count x entire. So, in that case you need to compute that how many alanine exists in the total protein. Now in the total protein definitely 3 plus number of alanine will be present because 3 is already in my deleted region. If I assume that say 5 number of alanine, so 5 alanines are present

in the entire and if the length of my protein is 75 then 3 by 10 divided by 5 by 10 is going to be the propensity for alanine at the deleted region.

Now, it may possible since you are deleting, say 10 amino acids or not always more than 20 amino acids. So, it may be possible that one particular deletion in one region, so several amino acids are not present at all. So, if they are not present, so count Del_stretch is going to be 0. And since it is 0, forget about everything else, so this will be 0. So, propensity is going to be 0.

Now, during the calculation of this propensity, be careful about one thing. Sometimes it may happen; in one amino acid sequence or in one protein sequence, one particular amino acid is not present at all in the entire structure itself. If it is not present, so in order to calculate that propensity this term is going to be 0. This will be 0. All will be 0. So, division of 0 by 0 that may throw some exception in your implementation. So, while you are implementing this propensity, so be careful about out this exception thrown because of the division by 0. Otherwise it is fine, and you can go ahead and compute all the features which are deletion site specific features.

(Refer Slide Time: 22:59)

MPD site features- Environment compatibility

- Total number of residues in the left (before Residue X) and right (after Residue Y) sub-units
- Inter-sub-unit Hydrogen bond, Salt bridges and Disulphide bonds
- Mean and standard deviation of WCN of the deleted residues
- Mean and standard deviation of WHbCN of the deleted residues

Pralay Mitra

Next, environment compatibility, total number of residues in the left before residue X and right after Residue Y subunits; Inter-sub-unit hydrogen bond, salt bridge and disulfide bond, mean and standard deviation of weighted Contact Number, that equation we presented during single point deletion and also I mentioned that some of the features of single point deletion will be used for

multi-point deletion. But additional stuff are also required. Mean and standard deviation of WHbCN of the deleted residue that is also defined in single point deletion.

(Refer Slide Time: 23:43)

MPD site features- Fold level attributes

$$\Delta\Delta G_{Folding}^{Del_stretch} = \Delta G_{Folding}^{Entire} - (\Delta G_{Folding}^{Left_subunit} + \Delta G_{Folding}^{Right_subunit})$$

Total FoldX energy (and 15 individual features) of the protein ($\Delta G_{Folding}^{Entire}$), its left sub-unit ($\Delta G_{Folding}^{Left_subunit}$) and its right sub-unit ($\Delta G_{Folding}^{Right_subunit}$)

Pralay Mitra

Now, MPD site features like fold level attributes. So, delta delta G or change in Gibbs Free Energy Del_stretch folding equals to delta G entire folding minus delta G left subunit folding plus delta G right subunit folding. So, total FoldX energy E and 15 individual features of the protein delta G entire folding, its left subunit and its right subunit is, using that one I will go for fold level attributes.

(Refer Slide Time: 24:20)

MPD site features- Evolutionary information

Considering MSA on identifying structural homologs using TM-Align

- Normalized Gap Count

$$NGC_{Del_stretch} = \frac{\sum_{i=1}^{Del_{end}} \sum_{j=Del_{start}}^{Del_{end}} K_{ij}}{N * (Del_{end} - Del_{start} + 1)}$$
 where $K_{ij} = \begin{cases} 1, MSA_{ij} \text{ is a gap} \\ 0, \text{ otherwise} \end{cases}$
- Similarly NGC_{Entire} assuming $Del_{start} = 1$ and $Del_{end} = \text{length of the protein}$
- Category Match Count

$$CMC_{Del_stretch} = \frac{\sum_{i=1}^{Del_{end}} \sum_{j=Del_{start}}^{Del_{end}} K_{ij}}{N * (Del_{end} - Del_{start} + 1)}$$
 where $K_{ij} = \begin{cases} 1, \text{ if } type(MSA_{ij}) = type(AA_j) \\ 0, \text{ otherwise} \end{cases}$
- Similarly CMC_{Entire} assuming $Del_{start} = 1$ and $Del_{end} = \text{length of the protein}$
- Support Gap Count $SGC_{Del_stretch} = \frac{\sum_{i=1}^{Del_{end}} \sum_{j=Del_{start}}^{Del_{end}} K_{ij}}{(Del_{end} - Del_{start} + 1)}$ where $K_{ij} = \begin{cases} 1, MSA_{ij} \text{ is a gap} \\ 0, \text{ otherwise} \end{cases}$
- Number of structural homologs

Pralay Mitra

So, these are the other evolutionary information. So, equations are given to you. So, if you want if you can easily calculate that one but considering MSA on identifying structural homologs using TM-align then you go for calculating these features.

(Refer Slide Time: 24:40)

The Positive Unlabeled learning algorithm

Algorithm 19: PROFOUNDP,U

Input: Lists of feature vectors P and U pertaining to the positive and unlabeled instances

Output: The PROFOUND classifier

- $X[1:P_{len}] \leftarrow P[1:P_{len}], X[P_{len}+1:P_{len}+U_{len}] \leftarrow U[1:U_{len}]$
- $y[1:P_{len}] \leftarrow 1, y[P_{len}+1:P_{len}+U_{len}] \leftarrow 0$
- Initial classifier** $\leftarrow RFC(X, y)$ Variables in bold denote entire array
- $c \leftarrow$ Estimate **initial classifier**, $5fold$
- $P_{U[1:U_{len}]}$ \leftarrow predict **probability**(**initial classifier**, $U[1:U_{len}]$)
- $(w_{pos}[1:U_{len}], w_{neg}[1:U_{len}]) \leftarrow$ calculate **weight**($P_{U[1:U_{len}]}, c$)
- $X_{train}[1:P_{len}+U_{len}] \leftarrow X[1:P_{len}+U_{len}], X_{train}[P_{len}+U_{len}+1:P_{len}+2U_{len}] \leftarrow U[1:U_{len}]$
- $y_{train}[1:P_{len}+U_{len}] \leftarrow 1, y_{train}[P_{len}+U_{len}+1:P_{len}+2U_{len}] \leftarrow 0$
- $weight[1:P_{len}] \leftarrow 1, weight[P_{len}+1:P_{len}+U_{len}] \leftarrow w_{pos}[1:U_{len}], weight[P_{len}+U_{len}+1:P_{len}+2U_{len}] \leftarrow w_{neg}[1:U_{len}]$
- PROFOUND classifier** $\leftarrow RFC(X_{train}, y_{train}, weight)$

Pralay Mitra

Once you will calculate this features then this is your positive unlabelled learning algorithm. So, where input is list of feature vectors P and U pertaining to the positive and unlabeled instances and output is the PROFOUND classifier. So, the name was coined as PROFOUND which will take P, U as input.

Now the algorithm says for X, 1 to, sorry, for X 1 colon P length, then, so this is the P length will be X P length, then X P length plus 1, P length plus U length. So, 1 through U length up to, so that is the U length. So, P is going to be from 1 to P length, and P length to U length will be basically U length. So, they are placed side by side, that vectors. So, list of feature vectors I got for P and U.

So, first it is P a feature vector followed by U feature vector. So, here after that s going to be s, 1 P length. Then P length plus 1, length U, plus U length that is 0. So, initial classifier is variable in bold denote entire array. So, Random Forest classifier is given here. Then Estimate c using 5-fold classification, then predict probability using initial classifiers and the unlabeled one then calculate weight, then you train that one, and then finally PROFOUND classifier is the Random Forest classifier with X_train, Y_train and weight. Now these variables in bold; so here it indicates that it is the entire array.

(Refer Slide Time: 26:38)

PROFOUND- Foldability classification

	MPD at a loop region		MPD at a non-loop region	
	recall (%)	FR (%)	recall (%)	FR (%)
Without evolutionary features				
Positive: 87 Unlabeled: 4350 Positive: 66 Unlabeled: 3300				
5-fold CV	79.3	15.4	84.4	21.9
10-fold CV	82.2	14.8	85.2	21.5
With (without) evolutionary features*				
Positive: 76 Unlabeled: 3060 Positive: 55 Unlabeled: 2640				
5-fold CV	79.0 (76.6)	14.0 (16.4)	86.4 (83.8)	21.9 (22.3)
10-fold CV	79.8 (78.6)	13.7 (16.2)	88.1 (84.8)	20.9 (22.0)

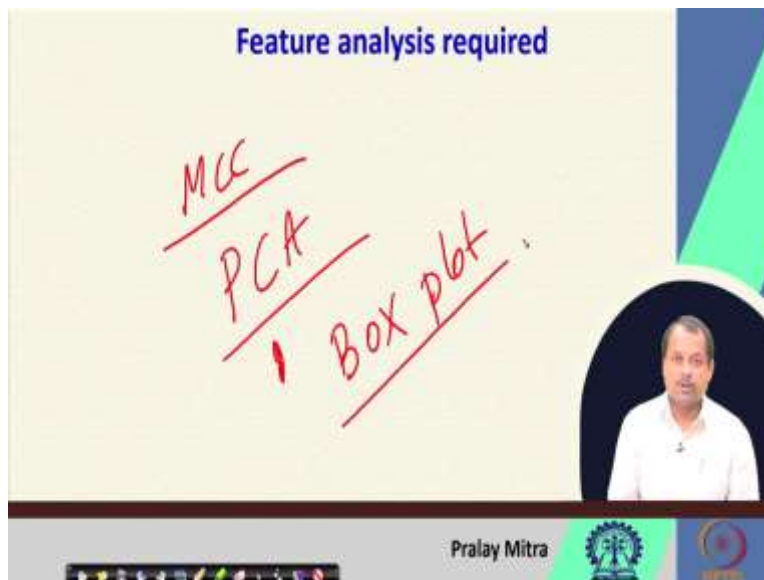
Banerjee et al. (2020) J. of Chemical Information and Modeling 60(12):6679-6690 Pralay Mitra

Now, with this algorithm basically we go for foldability classifications. So, we go for 5-fold cross validation. So, this cross validation we discussed in detail which indicates that from the entire pool, we will pick randomly, so a number of instances, and if it is 5-fold which means 5 buckets are there, and we will put in the buckets. Then 4 buckets will be considered for training and one for testing. In case of 10 fold cross validation 9 buckets will be used for, say training and one for the testing, that way.

Now, with or without evolutionary features you will see that the accuracy is given here. The accuracy is good but it is not that much impressive. But it is good with the 10-fold cross validation. However, given the scenario that multi-point deletion does not have any database, any features set and no other method to compare, so this actually is quite impressive but there is a scope of improvement.

But we are almost limited by the fact that at the dataset level we cannot do much, at the features set definitely we can contribute and also you can modify actually the algorithm. So, instead of P, U another machine language algorithm you can use and you can come up with some better accuracy for this foldability classification.

(Refer Slide Time: 28:17)



Of course one thing you have to keep in your mind while working with this feature list and the machine learning technique. Once you will have your feature then you have to go for feature level analysis. So, in case of predicting the biological assembly from the crystal lattice we mentioned that Matthews correlation Coefficient, in short MCC that can be used for analyzing the features. Or you can go for Principal Component Analysis or PCA in short, for analyzing the features. Or you can go for box plot in order to analyze the features. But whatever method it may be you have to analyze the features.

The primary reason is that your feature set should not be selected in such a way that it will bias result. And there should not be any correlation. So, if it is correlated then it may not give you the

required amount of result. So, that way, for feature, when you are developing some algorithm, specifically this machine learning algorithm, so two things you have to keep in your mind. The dataset must be balanced as much as possible.

In our case it was not balanced. That is why we used ADASYN. So, in that algorithm ADASYN was used. In another case, so when it is not, say dataset is balanced then if you are working with the feature, then feature level analysis is also required. So, with that let us stop here. Thank you very much.