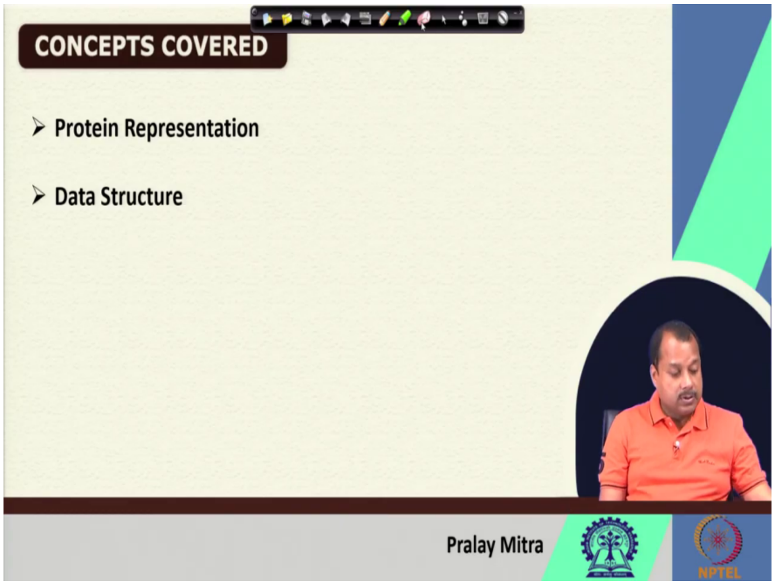
Algorithms for Protein Modelling and Engineering
Professor Pralay Mitra
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur
Lecture 06
Representation and Data Structure

Welcome back to the course of algorithms for protein modelling and engineering. We are starting the second module algorithmic techniques for modelling. It is a very long module. So, we shall break into different lectures and in different weeks. In this 6th lecture, we plan to introduce you to protein representation and the data structure which will be relevant for algorithm development.

(Refer Slide Time: 00:43)

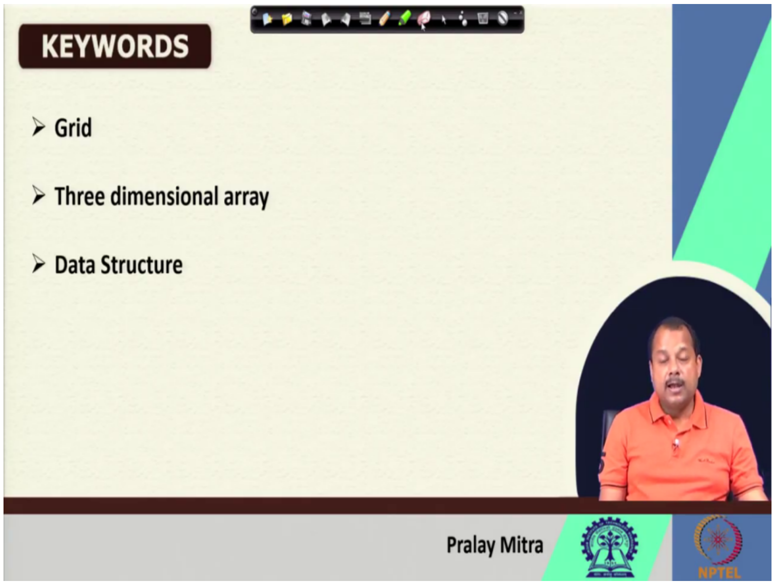


CONCEPTS COVERED

- Protein Representation
- Data Structure

The concept I would like to cover here is protein representation. Several times, you will encounter a protein representation topic, because, based upon the kind of algorithm or the application, the representation may be different. We understand that when we discuss relevant data structure.

(Refer Slide Time: 01:13)



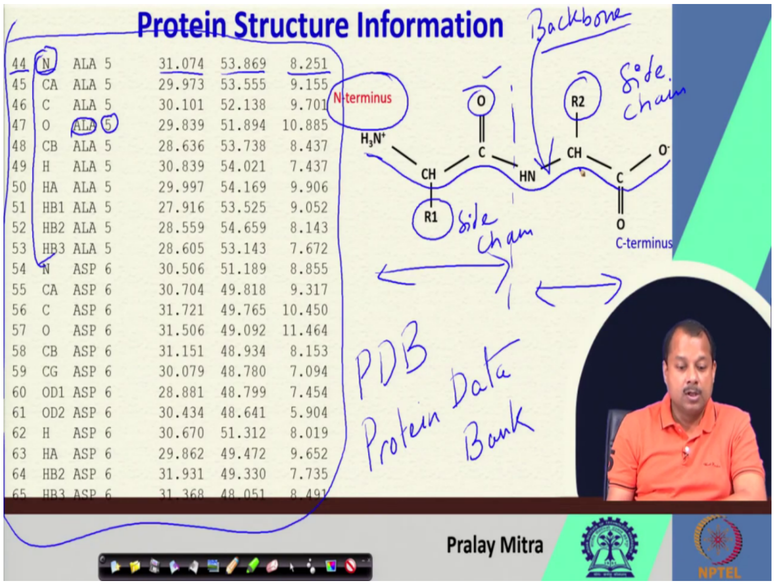
KEYWORDS

- Grid
- Three dimensional array
- Data Structure

Pralay Mitra

As keywords, we shall discuss the grid and when it is grid then you can understand that we plan to digitize the protein molecule to a three-dimensional array. This along with the data structure is also required for us.

(Refer Slide Time: 01:33)



Protein Structure Information

```
44 N   ALA 5     31.074 53.869  8.251
45 CA  ALA 5     29.973 53.555  9.155
46 C   ALA 5     30.101 52.138  9.701
47 O   ALA 5     29.839 51.894 10.885
48 CB  ALA 5     28.636 53.738  8.437
49 H   ALA 5     30.839 54.021  7.437
50 HA  ALA 5     29.997 54.169  9.906
51 HB1 ALA 5     27.916 53.525  9.052
52 HB2 ALA 5     28.559 54.659  8.143
53 HB3 ALA 5     28.605 53.143  7.672
54 N   ASP 6     30.506 51.189  8.855
55 CA  ASP 6     30.704 49.818  9.317
56 C   ASP 6     31.721 49.765 10.450
57 O   ASP 6     31.506 49.092 11.464
58 CB  ASP 6     31.151 48.934  8.153
59 CG  ASP 6     30.079 48.780  7.094
60 OD1 ASP 6     28.881 48.799  7.454
61 OD2 ASP 6     30.434 48.641  5.904
62 H   ASP 6     30.670 51.312  8.019
63 HA  ASP 6     29.862 49.472  9.652
64 HB2 ASP 6     31.931 49.330  7.735
65 HB3 ASP 6     31.368 48.051  8.491
```

PDB
Protein Data Bank

Pralay Mitra

In the last week, we concluded upto the protein structure information. Let me take it down. Now, you are familiar with this protein structure information. On the right-hand side, the protein sequence and the structure information is shown to you. Here, this is part of one protein and this is part of another protein. Now, when these two amino acids are connected through a covalent bond (also called a peptide bond) then they will form the protein sequence.

In this case, only two amino acids are considered and hence, it is a di-peptide. It can keep on going and that way, there may be hundreds of or thousands of amino acids. Generally, we restrict to a few hundred only for our computational purposes. Definitely, that can be extended for any number based upon what is the protein, what is its size.

Now, when we will store this atom information, then that information will store its coordinate, atom type, an atom is a part of which residue or the amino acid (We shall use residue and amino acid interchangeably), the location of the residue. The location of the residue indicates if I start from the N terminal region on the left-hand side inside that protein sequence and the protein structure, what is the serial number of a particular atom by which I can probably identify or locate that particular atom and the other information.

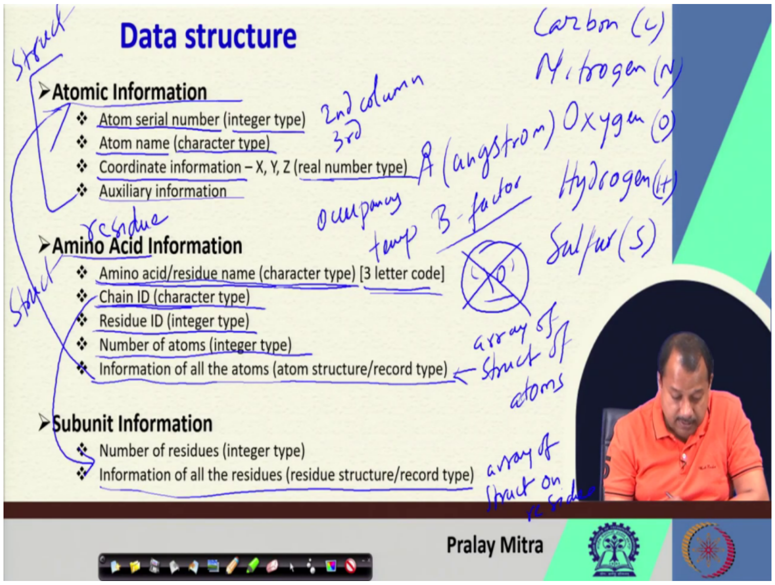
Other information is not included here because that is not relevant to our discussion right now. During the discussion on the implementational details, I shall show you that which information is present there and from there how to strip or how to pre-process the data, so that we will get all the data relevant information.

It is needless to mention that this particular information which is represented here is taken from PDB - Protein Data Bank that we have discussed extensively on the last week. Now, to process this Protein Data Bank information for any protein modelling or protein engineering, I need to store the structural information somewhere.

There may be several ways to store that information. Keeping an eye towards its application, I mean where it will be applied the information storing will also vary. Now what you cannot deny is that corresponding to each amino acid you need to store the name of the amino acid, the amino acid number, the number of atoms representing amino acid, corresponding to each atom X-, Y- and Z- coordinate, the serial number of the atom, whether it is part of the backbone or part of the side chain. I hope by this time you know that when I say backbone, then basically I have to follow this one - this is my backbone.

Since there are only two amino acids, so, I am extending from the N-terminus to the C-terminus. In between, this is the backbone. What is the side chain? This is side chain, this is side chain and this one I am calling as because this backbone determines the fold of the protein structure. Primarily, this information I have to store.

(Refer Slide Time: 07:02)



Data structure

➢ Atomic Information
- ❖ Atom serial number (integer type)
- ❖ Atom name (character type)
- ❖ Coordinate information – X, Y, Z (real number type)
- ❖ Auxiliary information

➢ Amino Acid Information
- ❖ Amino acid/residue name (character type) [3 letter code]
- ❖ Chain ID (character type)
- ❖ Residue ID (integer type)
- ❖ Number of atoms (integer type)
- ❖ Information of all the atoms (atom structure/record type)

➢ Subunit Information
- ❖ Number of residues (integer type)
- ❖ Information of all the residues (residue structure/record type)

Pralay Mitra

Now, I shall show you what is the preferred data structure for me, so that I can store that information. Regarding a data structure or record, I have to declare which will store the atomic information. I have the atom serial number that I demonstrated to you it will be in the second column of the PDB file.

After the atom it is the residue information then chain information after that the coordinate information will start. In coordinate information, the first column is the X, the next column is the Y and the third column is the Z and apart from that one, there is some auxiliary information. Please note that the dimension of the coordinate is in angstrom and is denoted as Å.

What kind of auxiliary information is there? One among them is occupancy. What is the occupancy of the atom? Usually, we get the protein structure at the atomic level resolution utilizing some experimental techniques, for example, X-ray crystallography or nuclear magnetic resonance etcetera. When is it an experiment, then what is the probability of getting a particular atom in a particular residue? That probability value provides the occupancy information. Usually, it varies from some non-zero value to 1.0, where 1.0 indicates that with 100% probability I am getting the atom. Something less than 1.0 indicates that experimentally it is not certain.

After occupancy information is the temperature factor or B factor. In X-ray crystallography or for experimental data, that auxiliary information may not be relevant for all the algorithmic techniques. But if you wish to use that one for calculating some scoring value or some other calculations, then you may store that one otherwise you can ignore it. Please note that corresponding to each information some storage or memory is required and nothing is free so, if it is not required to store unnecessary information.

After atomic information is amino acid information. As I mentioned that I shall use this amino acid and residue interchangeably. So, I can consider also the residue information. What is there in the residue information? Amino acid or residue name is a character type. Sorry, I forget to mention that in the atomic information, the atom serial number is an integer type that can assume a negative, 0, or positive number. Please remember this.

I understand that you may think what is the significance of a negative atom number? In protein databank, you may encounter some protein structures where the atom serial number starts with negative indicating that perhaps my area of interest or the atom from which the protein sequence starts will be numbered starting from 1, but if some additional information is also appended with that particular protein sequence just because of the experimental purposes, then that information will be stored with the negative atomic serial number. There is no problem with it as far as the information is in PDB. Nevertheless, we shall see shortly that a negative atom serial number might be a problem for computational processing. Another concern is the atom name which is of character type.

Amino acids usually contain carbon C, nitrogen N, oxygen O, hydrogen, sulphur. These are the atoms and their names C, N, O, H, S will be in PDB. Regarding positional information, whether it is part of the backbone or side chain will also be stored. That is why in the atom name although it is a character type, there will not be a single character there may be multiple characters. For example, if it is N, which means it is the nitrogen at the backbone, if it is O that means, it is the oxygen, part of the carboxylic group at the backbone, if it is CA that means, it is the C alpha atom. Similarly, CB means the carbon of the sidechain at the beta position. Beta position means the carbon at the side chain which is forming the covalent bond with the C alpha atom. In summary, an atom is a character type that can be multiple characters. Regarding coordinate information, they are real number types. And auxiliary information may be a character, a real number, or an integer, based upon the kind of information.

Coming back to amino acid information, we need to store the name of the amino acid that is denoted by three-letter characters in PDB. At the introduction of protein, I introduced to you the full names, three-letter codes and a single character code (taken from the English alphabets) of 20 amino acids. In PDB, it is the three-letter code that has been used.

While storing one structure information either you can store three-letter code directly or you can have one separate function that will read three-letter code, will convert to a single character code and will store the single character code. Please note that storing a three-letter code requires more memory storage than storing a single character code. If I am thinking of implementing in C or C++ then this three-letter code will be of character array terminated by a ‘\0’ which means four characters are required compared to only one character for storing a single character code. This will save some storage space. After your processing, if you wish to write that information back, or you need to print the three-letter code for the amino acid instead of the single character you revert it. Thus, you need to have two functions one for converting from 3 to 1 letter code another for converting from 1 to 3 letter code and both will run only once at the beginning and at the end. Considering the reduced storage requirement you will not mind having those two functions in place and executing those at the beginning and at the end, respectively. Please note that these two functions can be reused for several purposes. Also, you should not mind writing such small functions and declaring those as library functions.

Next, in the amino acid information, you need to store the chain ID as the character type. Even if you find that it is a digit 0, 1, 2, 3, 4 but always store that as a character. It is character 0, character 1, character 2 etcetera. Because that character will include A, B, C along with that 0, 1, 2, small a capital A everything. Next is residue ID similar to atom serial number. Residue ID or residue serial number is also required to store an integer type. Similar to atom numbers, you may encounter some situations where the residue number is negative. This also suggests that after reading the PDB probably you have to parse or you have to rewrite that PDB or you have to convert or modify that PDB for your applications.

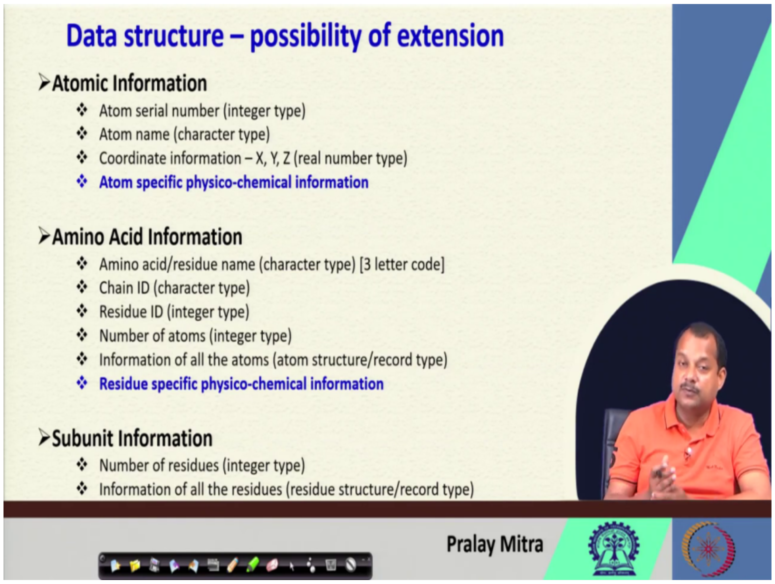
Software like VMD reads the PDB file, but it does not rely on the atom serial number or the residue ID which is provided in the PDB. Rather it renumbers on its own starting from 1.

The number of atoms is crucial information that must be stored with the amino acid. It is an integer type and it is a non-zero non-negative integer type that indicates how many atoms are present in that particular amino acid. Mostly this information varies from one amino acid to another amino acid unless otherwise, some atoms are missing in the PDB file. Remember that there may be situations where hydrogen atoms are not reported in the PDB, there may be situations some atoms may be missing due to experimental procedure and hence the number of atoms may vary. Despite it being a constant integer, say for alanine the number of atoms will be the same - it is true, but for the input data it may not be the case always. So, it is better to store the actual information. Next array or list of atomic information that will come from here. List up this record or this structure, atomic information will also be part of amino acid information.

If you declare these as in C programming language, then one structure and here this is my another structure. Then this is going to be the structure of atoms or I can write an array of the structure of atoms. The third one you need to store is the subunit information. I mentioned in the introduction class that a protein is composed of one or multiple chains and when there are multiple chains each chain can be considered as one subunit; one connected component where there is a peptide bond and where from any amino acid you can traverse to another amino acid. Hence, that is the connected part.

Corresponding to each connected part there is one ID that ID is called as that chain ID that's why that connected part is also called a protein chain. The protein chain or chain ID you need to store, which is also mentioned here as chain ID. This needs to be stored along with the number of residues - a nonzero non-negative positive integer. Again information of all the residues structure or record types will be an array of structures on residues. You need to store that information too.

(Refer Slide Time: 22:43)



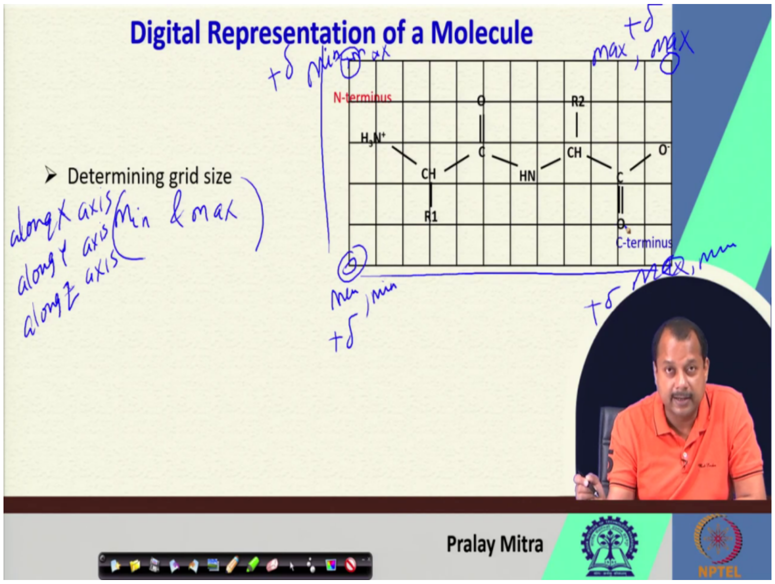
Data structure – possibility of extension

- **Atomic Information**
  - Atom serial number (integer type)
  - Atom name (character type)
  - Coordinate information – X, Y, Z (real number type)
  - **Atom specific physico-chemical information**
- **Amino Acid Information**
  - Amino acid/residue name (character type) [3 letter code]
  - Chain ID (character type)
  - Residue ID (integer type)
  - Number of atoms (integer type)
  - Information of all the atoms (atom structure/record type)
  - **Residue specific physico-chemical information**
- **Subunit Information**
  - Number of residues (integer type)
  - Information of all the residues (residue structure/record type)

Next, we wish to move to that digital world for ease of calculation. Because the coordinate information is a real number. During the calculation dealing with the real number is more computationally costly than dealing with integer numbers. Thus, we are planning to digitize the protein molecule. Along with the digitized information, we need to store some auxiliary information as part of the atomic information like atomic charge which is not present in the protein databank. This information can be pulled from some literature or the force-field of some software. This auxiliary information we can consider as part of the physical chemical or physiochemical information is physical as well as chemical information like the charge, the atomic mass, the hydrophobicity score of the residue.

Hence, those features or those properties relate to each amino acid or each atom we wish to store. Also, that information we would like to store in the structure. Right now I am not going to name those it will appear when it will be required. Anyway, I am giving a heads-up to mention that although the black colour information is the member of the structure of atomic information, residue information and subunit information, sometimes additional or auxiliary information may be required. Therefore, an option to store that one is required.

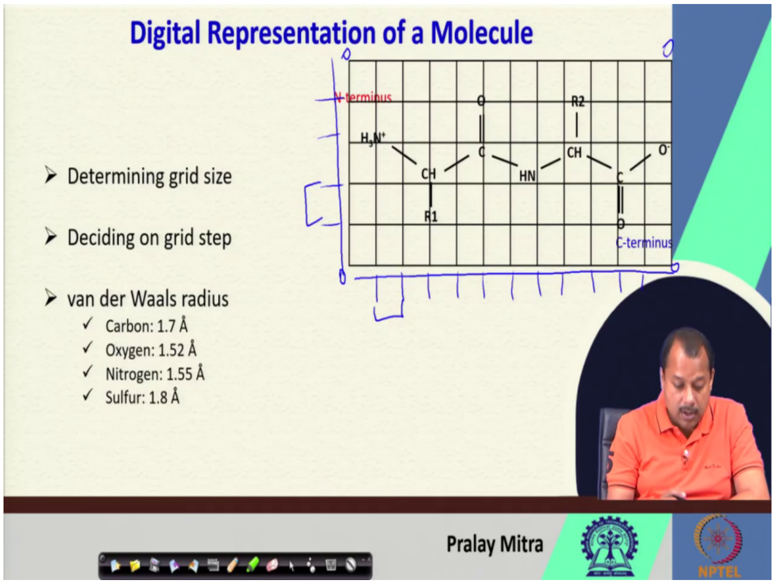
(Refer Slide Time: 25:29)



Digital Representation of a Molecule

- Determining grid size

along X axis
along Y axis
along Z axis (Min & max)



Digital Representation of a Molecule

- Determining grid size
- Deciding on grid step
- van der Waals radius
  - ✓ Carbon: 1.7 Å
  - ✓ Oxygen: 1.52 Å
  - ✓ Nitrogen: 1.55 Å
  - ✓ Sulfur: 1.8 Å

Digital representation of a molecule says that I wish to put that molecule in some grid. In this case, you can see that although in reality, the molecule is a three-dimensional structure, here I am projecting it on a two-dimension and that is why I draw one two-dimensional grid. The molecule is inside this grid. When it is inside this grid along the X- and Y- axis I need to decide what would be the grid size for digitizing the protein. It is very simple. What do you need to calculate is in two dimension is minimum and maximum along X- and Y-axis. If it is three dimensional, add one more axis along Z and calculate minimum and maximum.

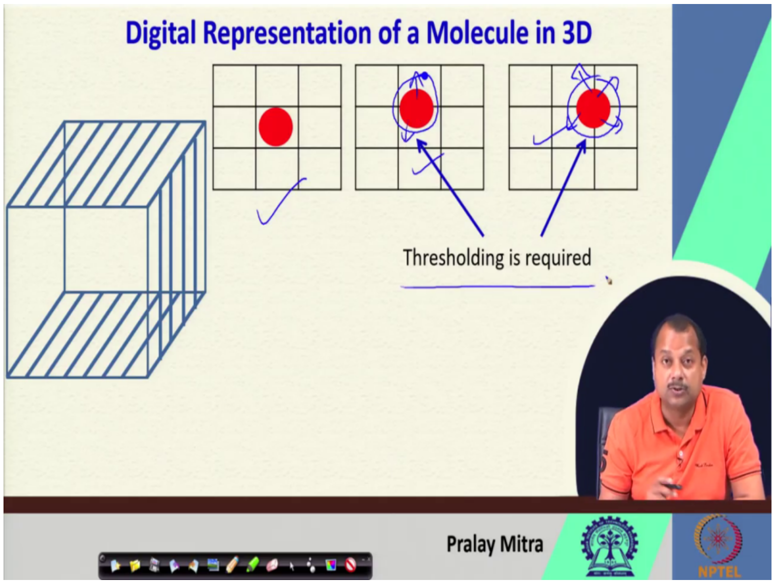
This is my minimum point, this is my maximum point, this is my maximum point, this is my minimum point. This is the min-max this is the max-max, this is the min-min and this is the max-min. Along the X-axis is minimum, maximum. Along the Y-axis is the minimum maximum. Here X- and Y- both are maximum.

After reading the protein databank file, I store that information in the structure or the record that I mentioned. After storing coordinate information, what do I have to do from each atom? I have to calculate what is the minimum X what is the maximum X what is the minimum Y what is the maximum Y what is the minimum Z, what is the maximum Z. Along with that I need to give some offsets of delta (a few grids) in all directions. After that I shall decide on the grid size, once I decide the grid size, then I fix the size of this rectangle where the outside lines, the inner lines are the number of rows and columns. If I consider that it is a kind of matrix, then the cell size will be decided by the grid step.

I decided on the grid size, which means, this point, this point, this point, and this point is decided. Now, this entire line will be divided or partitioned and also this one will be divided or partitioned. I have to decide on this small step along the X-axis and along the Y-axis. If there is Z-axis, along the Z-axis also, I have to decide on the grid step by noting the atom size.

Despite the atoms are not a hard-sphere, for our purposes, we shall consider a ball and stick model and we will consider an atom as a hard-sphere whose radius is given by the van der Waals radius. Likewise, carbon has 1.7 Å, oxygen 1.52 Å, nitrogen 1.55 Å, sulphur 1.8 Å, all are in angstrom. Hydrogen is around 1.02 Å. There may be some small variation in that based upon different experimental techniques. Grossly these are the values. Keeping those things in mind probably, I can decide on what will be the grid size.

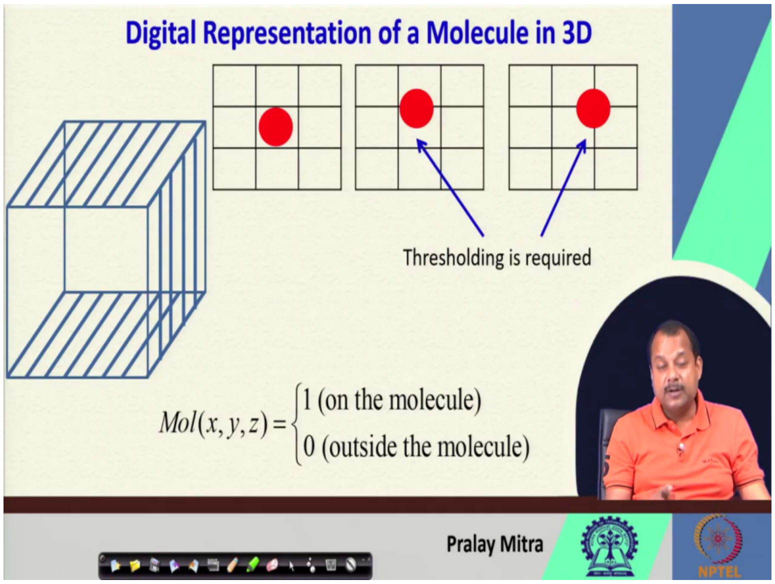
(Refer Slide Time: 30:27)



Digital Representation of a Molecule in 3D

Thresholding is required

Pralay Mitra



Digital Representation of a Molecule in 3D

Thresholding is required

$$Mol(x,y,z)=\begin{cases}1\text{ (on the molecule)}\\0\text{ (outside the molecule)}\end{cases}$$

Pralay Mitra

Also, we have to consider three different situations – (i) when one particular atom is inside the cell rightly placed as you can see that one, (ii) it is covering two adjacent cell positions, (iii) when it is covering four adjacent cell positions. This is in two dimensions. If I map it to three dimensions it will be something here on the blue colour left-hand side. For the first case, like this one, I do not have absolutely any problem. But for these two cases, I have a small problem. What is that problem? I need a threshold - threshold of what? After placing this red circle and this red circle, I have to decide whether it belongs to this cell, this cell, this cell, this cell, this cell, this cell or not. Thus, I have to go for some thresholding. After the thresholding I declare one function, that function says that *Mol(x,y,z)* equals 1, if it is on the molecule, it is 0 if it is outside the molecule. Now, you will see that this particular function will make your life very easy for calculating the surface overlap that we shall discuss in the next lecture. Thank you.