Introduction to Bioinformatics

Protein Structure Informatics: Bio.PDB

Janyl Jumadinova
April 14, 2016
Proteins...

a linear sequence of amino-acids, after transcription from DNA, and translation from mRNA
Proteins...

3-D molecules that interact with other (biological) molecules to carry out biological functions
Protein Data Bank (PDB)

- Repository of the 3-D conformation(s) / structure of proteins.
Protein Data Bank (PDB)

- Repository of the 3-D conformation(s) / structure of proteins.
- The result of laborious and expensive experiments using X-ray crystallography and/or nuclear magnetic resonance (NMR).
  - \((x,y,z)\) position of every atom of every amino-acid
Visualizations
Biopython Bio.PDB

- Parser for PDB format files
- Navigate structure and answer atom-atom distance/angle questions
Biopython Bio.PDB

- Parser for PDB format files
- Navigate structure and answer atom-atom distance/angle questions

1. Create a PDBParser object
   
   ```python
   parser = PDBParser()
   ```
Biopython Bio.PDB

- Parser for PDB format files
- Navigate structure and answer atom-atom distance/angle questions

1. Create a PDBParser object
   ```python
   parser = PDBParser()
   ```

2. Create a structure object from a PDB file
   ```python
   structure = parser.get_structure('1g59', '1g59.pdb')
   ```
Can access header information

```python
resolution = structure.header['resolution']
keywords = structure.header['keywords']
```

The available keys are `name`, `head`, `deposition_date`, `release_date`, `structure_method`, `resolution`, `structure_reference` (maps to a list of references), `journal_reference`, `author` and `compound` (maps to a dictionary with various information about the crystallized compound)
The **Structure** object follows the so-called SMCRA (Structure/Model/Chain/Residue/Atom) architecture.
The **Structure** object follows the so-called SMCRA (Structure/Model/Chain/Residue/Atom) architecture.

- Structure (PDB File) >> Model >> Chain >> Residue >> Atom >> (x,y,z) coordinates
- SMCRA representation mirrors PDB format

Biopython Bio.PDB

[8/17]
### PDB File

<table>
<thead>
<tr>
<th>atom number</th>
<th>name (C, H, O, N...) &amp; position (A=α, B=β, G=γ)</th>
<th>X, Y and Z coordinates</th>
<th>occupancy (fraction in each alternative structure)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATOM 89</td>
<td>N VAL A 11</td>
<td>14.377 10.760 18.401</td>
<td>1.00 27.95 N</td>
</tr>
<tr>
<td>ATOM 90</td>
<td>CA VAL A 11</td>
<td>15.403 11.313 19.262</td>
<td>1.00 26.76 C</td>
</tr>
<tr>
<td>ATOM 91</td>
<td>C VAL A 11</td>
<td>16.766 10.870 18.760</td>
<td>1.00 28.04 C</td>
</tr>
<tr>
<td>ATOM 92</td>
<td>O VAL A 11</td>
<td>16.871 9.951 17.943</td>
<td>1.00 26.79 O</td>
</tr>
<tr>
<td>ATOM 93</td>
<td>CB VAL A 11</td>
<td>15.221 10.851 20.717</td>
<td>1.00 26.79 C</td>
</tr>
<tr>
<td>ATOM 94</td>
<td>CG1 VAL A 11</td>
<td>13.841 11.273 21.223</td>
<td>1.00 26.42 C</td>
</tr>
<tr>
<td>ATOM 95</td>
<td>CG2 VAL A 11</td>
<td>15.401 9.348 20.810</td>
<td>1.00 25.99 C</td>
</tr>
<tr>
<td>ATOM 96</td>
<td>N THR A 12</td>
<td>17.804 11.535 19.245</td>
<td>1.00 27.77 N</td>
</tr>
<tr>
<td>ATOM 97</td>
<td>CA THR A 12</td>
<td>19.163 11.215 18.845</td>
<td>1.00 28.45 C</td>
</tr>
<tr>
<td>ATOM 98</td>
<td>C THR A 12</td>
<td>19.752 10.167 19.773</td>
<td>1.00 29.05 C</td>
</tr>
<tr>
<td>ATOM 99</td>
<td>O THR A 12</td>
<td>19.684 10.292 20.994</td>
<td>1.00 29.41 O</td>
</tr>
<tr>
<td>ATOM 100</td>
<td>CB THR A 12</td>
<td>20.057 12.475 18.867</td>
<td>1.00 29.38 C</td>
</tr>
<tr>
<td>ATOM 101</td>
<td>OG1 THR A 12</td>
<td>19.636 13.376 17.836</td>
<td>1.00 30.85 O</td>
</tr>
<tr>
<td>ATOM 102</td>
<td>CG2 THR A 12</td>
<td>21.521 12.105 18.648</td>
<td>1.00 29.38 C</td>
</tr>
</tbody>
</table>
Navigate Through the Structure

Iterate through all atoms of a structure:

```python
parser = PDBParser()
structure = parser.get_structure('1g59', '1g59.pdb')
for model in structure:
    for chain in model:
        for residue in chain:
            for atom in residue:
                print atom
```
Navigate Through the Structure

Iterate through all atoms of a structure:

```python
# Iterate over all atoms in a structure
for atom in structure.get_atoms():
    print atom
```

```python
# Iterate over all residues in a model
for residue in model.get_residues():
    print residue
```
Navigate Through the Structure

- Structures, models, chains, residues and atoms are called **Entities** in Biopython

- You can always get a **parent** Entity from a **child** Entity and vice versa

```
residue = atom.get_parent()
chain = residue.get_parent()
```
SMCRA Data-Model

- Each PDB file represents one “structure”
Each PDB file represents one “structure”

Each structure may contain many models
- In most cases there is only one model, index 0
Each PDB file represents one “structure”

Each structure may contain many models
- In most cases there is only one model, index 0

Each polypeptide (amino-acid sequence) is a “chain”
- A single-protein structure has one chain, “A”
- 1HPV is a dimer and has chains “A” and “B”
SMCRA Data-Model

```python
import Bio.PDB.PDBParser
import sys

# Use QUIET=True to avoid lots of warnings...
parser = Bio.PDB.PDBParser(QUIET=True)
structure = parser.get_structure("1HPV", "1HPV.pdb")
model = structure[0]

# This structure is a dimer with two chains
achain = model['A']
bchain = model['B']
```
- Chains are composed of amino-acid residues
  - Access by iteration, or by index
  - Residue “index” may not be sequence position
- Residues are composed of atoms:
  - Access by iteration or by atom name
  - ...except for H!
- Water molecules are also represented as atoms HOH residue name, het=“W”
```python
import Bio.PDB.PDBParser
import sys

# Use QUIET=True to avoid lots of warnings...
parser = Bio.PDB.PDBParser(QUIET=True)
structure = parser.get_structure("1HPV", "1HPV.pdb")
model = structure[0]

for chain in model:
    for residue in chain:
        for atom in residue:
            print chain, residue, atom, atom.get_coord()
```