PDB Autofill

Henry Lee, You-Hsin Chen, and Jenny Bennett

UNIVERSITY of WASHINGTON

Background

- **Our goal:** Classify the reason for missing electron densities of protein crystals in PDB by using a random forest model
- **Stretch:** Predict the missing density coordinates in proteins with a neural network model

Sequence Number

```
ATOM    437  CG2 VAL A  47
ATOM    438  H   VAL A  47
ATOM    439  N   GLY A  52
ATOM    440  CA  GLY A  52
```

```
REMARK 465 MISSING RESIDUES
REMARK 465 THE FOLLOWING RESIDUES WERE NOT LOCATED IN THE
REMARK 465 EXPERIMENT. (M=MODEL NUMBER; RES=RESIDUE NAME; C=CHAIN
REMARK 465 IDENTIFIER; SSSEQ=SEQUENCE NUMBER; I=INSERTION CODE.)
REMARK 465
REMARK 465   M RES C SSSEQI
REMARK 465     GLY A    48
REMARK 465     GLY A    49
REMARK 465     ILE A    50
REMARK 465     GLY A    51
REMARK 470
REMARK 470 MISSING ATOM
REMARK 470 THE FOLLOWING RESIDUES HAVE MISSING ATOMS (M=MODEL NUMBER;
REMARK 470 RES=RESIDUE NAME; C=CHAIN IDENTIFIER; SSEQ=SEQUENCE NUMBER;
REMARK 470 I=INSERTION CODE):
REMARK 470   M RES CSSEQI  ATOMS
REMARK 470     PHE A  53    CG   CD1  CD2  CE1  CE2  CZ
REMARK 470     LYS A  60    CG   CD   CE   NZ
```

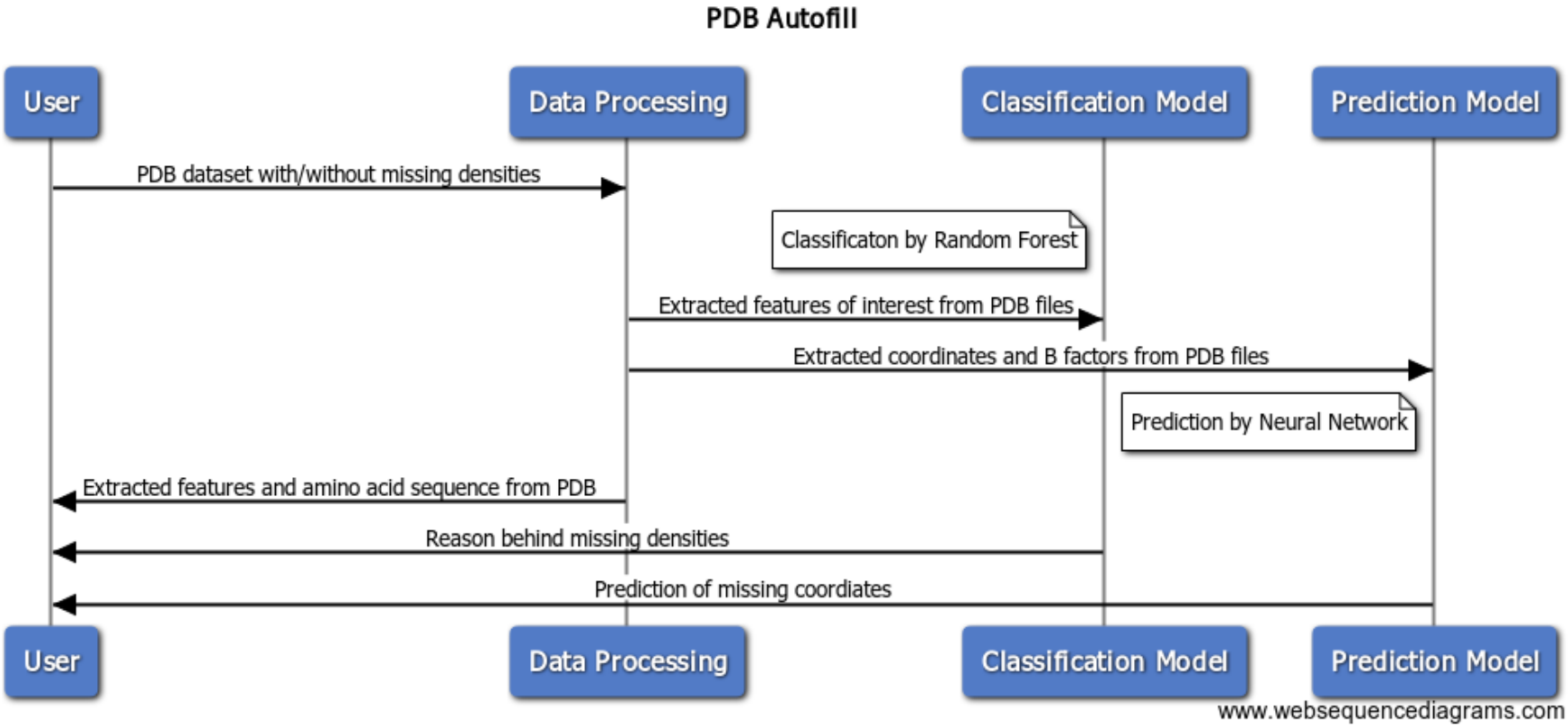
Use Cases

Alice is a computational scientist who will be using the tool to determine the expected structure of her proteins of interest. Alice will use this package to:

- **Classify why her proteins of interest have missing densities in the PDB.**
- **Predict the missing densities in her proteins of interest.**
- **Extract sequences and features in a DataFrame to perform further analysis on her own.**

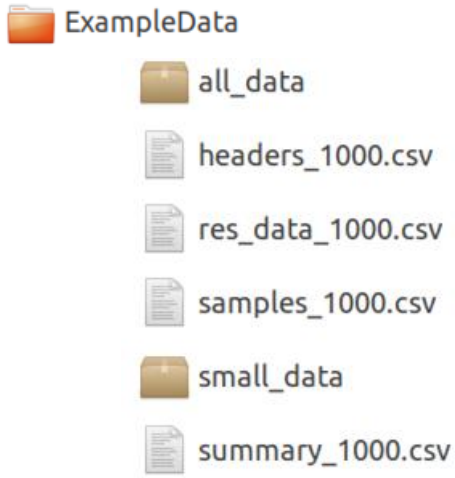
Joe is new to data science and is interested in learning more about PDB files. He will use the package to:

- **Download a sample dataset of PDB files.**
- **Extract interesting features from the PDB files.**
- **Classify and predict the missing densities that are not captured in the files.**


PDB Autofill
User
Data Processing
Classification Model
Prediction Model
PDB dataset with/without missing densities
Classificaton by Random Forest
Extracted features of interest from PDB files
Extracted coordinates and B factors from PDB files
Prediction by Neural Network
Extracted features and amino acid sequence from PDB
Reason behind missing densities
Prediction of missing coordiates
User
Data Processing
Classification Model
Prediction Model
www.websequencediagrams.com

Downloading Data

- Application:
  1) Users can download data we used in example and do the following PDB analysis.
  2) The package will create a folder called ExampleData on the working directory and download two compressed .zip files which include 3000 .pdb and 20 .pdb files respectively and four .csv files which are our input files.



UNIVERSITY of WASHINGTON

Data Processing

- Application:
  1) Count how many .pdb files in the folder has missing residues.
  2) Create new dataset which contains 50% proteins w/ missing residues and 50% proteins w/o missing residues for training data in order to avoid imbalanced class of follow-up classified model construction

Name of proteins

Biopython glob

statistics

```
Total nuumber of samples: 2970
Samples have missing residues: 2422
Samples without missing residues:: 548
There are 81.55% samples have missing residue
```

Random pandas

	Protein
0	2y39
1	2o73
2	3d5m
3	1gey
4	4y79
...	...

Load pdb files we chose and go extracting information from them

UNIVERSITY of WASHINGTON

Data Processing

- Application :

  3) Extract useful information from .pdb files downloaded from Protein Data Bank

```
REMARK   2 RESOLUTION.    2.30 ANGSTROMS.
REMARK   3
REMARK   3 REFINEMENT.
REMARK   3   PROGRAM     : X-PLOR 3.851
REMARK   3   AUTHORS     : BRUNGER
REMARK   3
REMARK   3  DATA USED IN REFINEMENT.
REMARK   3   RESOLUTION RANGE HIGH (ANGSTROMS) : 2.30
REMARK   3   RESOLUTION RANGE LOW  (ANGSTROMS) : 8.00
REMARK   3   DATA CUTOFF            (SIGMA(F)) : 2.000
REMARK   3   DATA CUTOFF HIGH         (ABS(F)) : NULL
REMARK   3   DATA CUTOFF LOW          (ABS(F)) : NULL
REMARK   3   COMPLETENESS (WORKING+TEST)   (%) : 97.8
REMARK   3   NUMBER OF REFLECTIONS             : 16008
REMARK   3
```

```
ATOM      1  N   THR A   5     -12.150   6.886  19.540  1.00 44.37           N
ATOM      2  CA  THR A   5     -12.336   8.358  19.394  1.00 44.49           C
ATOM      3  C   THR A   5     -11.744   8.875  18.079  1.00 41.37           C
ATOM      4  O   THR A   5     -11.126   8.117  17.331  1.00 39.88           O
ATOM      5  CB  THR A   5     -11.707   9.123  20.579  1.00 48.31           C
ATOM      6  OG1 THR A   5     -11.988  10.525  20.447  1.00 54.98           O
ATOM      7  CG2 THR A   5     -10.193   8.902  20.628  1.00 49.10           C
ATOM      8  N   ILE A   6     -11.927  10.171  17.822  1.00 37.21           N
ATOM      9  CA  ILE A   6     -11.440  10.824  16.605  1.00 34.51           C
ATOM     10  C   ILE A   6     -10.004  10.463  16.251  1.00 32.78           C
ATOM     11  O   ILE A   6      -9.701  10.161  15.096  1.00 32.55           O
ATOM     12  CB  ILE A   6     -11.545  12.365  16.719  1.00 31.74           C
ATOM     13  CG1 ILE A   6     -12.995  12.777  16.976  1.00 33.50           C
```

PDB parser from BioPython

Dataset with some headers(resolution/statistics of b factor/has missing residue)

Protein	name	head	structure_method	resolution
2y39	ni-bound form of cupriavidus metallidurans ch...	metal binding protein	x-ray diffraction	1.41
2o73	structure of ohcu decarboxylase in complex wi...	lyase	x-ray diffraction	1.80
3d5m	crystal structure of hcv ns5b polymerase with...	transferase	x-ray diffraction	2.20
1gey	crystal structure of histidinol-phosphate ami...	transferase	x-ray diffraction	2.30
4y79	factor xa complex with gtc000406	hydrolase	x-ray diffraction	2.10

residues -> getting properties (sequence length/ hydrophobic…)

	2y39	2o73	3d5m	1gey	4y79	3gem	2z91	2jfk	3ueo	3qun
0	GLY	ASP	SER	THR	ILE	SER	GLN	GLN	GLY	SER
1	ASP	ILE	MET	ILE	VAL	ALA	LEU	SER	LEU	GLY
2	LEU	ASN	SER	THR	GLY	PRO	LEU	MET	PHE	LEU
3	HIS	VAL	TYR	ASP	GLY	ILE	GLU	ARG	SER	VAL
4	GLU	VAL	THR	LEU	GLN	LEU	SER	LEU	GLN	PRO

Combine -> training data

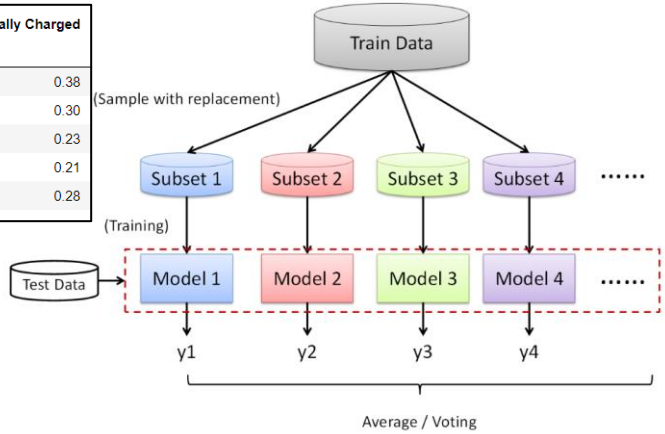
UNIVERSITY of WASHINGTON

Random Forest

- Application :

  1) Users can build up own random forest model, and list importance of each feature in the model and mean accuracy under 5-fold cross-validation.

Protein	has_missing_residues	Sequence Length	resolution	b_factor_gt50	b_factor_max	Electrically Charged
2y39	True	110.0	1.41	0	46.46	0.38
2o73	True	992.0	1.80	4	65.24	0.30
3d5m	True	1116.0	2.20	3225	107.96	0.23
1gey	True	335.0	2.30	190	87.44	0.21
4y79	True	287.0	2.10	282	98.34	0.28




```
1) b_factor_gt50                0.181701
2) Sequence Length              0.153594
3) b_factor_max                 0.146267
4) Electrically Charged         0.126673
5) resolution                   0.118797
6) Hydrophobic                  0.097705
7) Nonpolar Side Chains         0.088332
8) Special                      0.086930
Accuracy of the prediction in 5-fold cross-validation = 71.27%
```

In our project, we used 1000 proteins with these 8 features to build up Random Forest model (500 decision trees). The average accuracy is 71.27 %. To enhance the performance, users could try adding more useful features, data or stacking with a boosting model.

UNIVERSITY of WASHINGTON

Random Forest

- Application :
  2) Users who are going to do experiment can use RF model we have already built to test if it tends to get missing residues under this circumstance.
  3) Users can get most used features of nodes that each input data went through.

```
RF_model = joblib.load('RandomForest_model.pkl') #Load model
```

input data → Prediction: has missing residues (yes/no)

In each decision tree

```
# The path of single input data goes through the random forest model
feature_count_accum = []
for j, tree in enumerate(RF_model.estimators_):
    # matrix of nodes that input data go through(boolean)
    dense_matrix = tree.decision_path(x[150].reshape(1, -1)).todense()
    #transform to array
    dense_sample = np.array(dense_matrix)[0]
    # extract number of nodes that input data goes through
    node_position = np.where(dense_sample == 1)[0]
    feature_count = []
```

Matrix shows nodes that input data goes through (value = 1)

```
matrix([[1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
         0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
         0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
         1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
         0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1,
         0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
         0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
         0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
         0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

Transform to position (number) of nodes

```
for i in range(len(node_position)):
    number = node_position[i]
    feature_count.append(feature_name[tree.tree_.feature[number]])
feature_count_accum.extend(feature_count)
```

Which feature that the node is ruled by

UNIVERSITY of

```
Most used features of nodes that input data went through [('Nonpolar Side Chains', 1050), ('b_factor_max', 945), ('b_factor_gt50', 930), ('Sequence Length', 891), ('resolution', 833), ('Electrically Charged', 797), ('Hydrophobic', 645), ('Special', 597)]
```

Data Processing

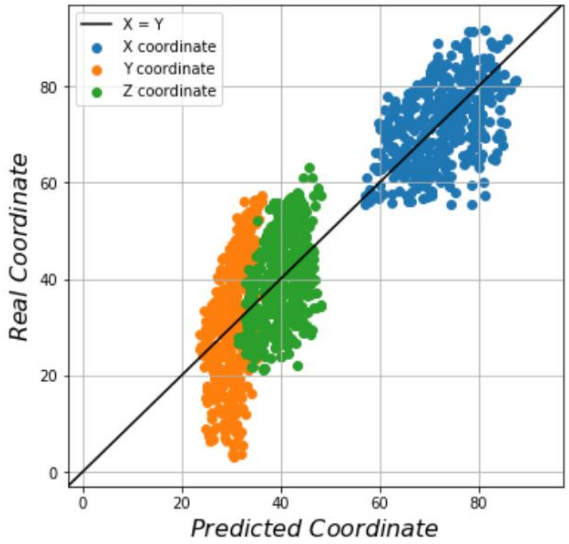
- Application:
  1) Creates a dataframe which contains the atom coordinates, the b values, the previous atom coordinates, the following atom coordinates.
  2) Creates a dataframe which contains missing residues, the corresponding protein chain and sequences for each protein

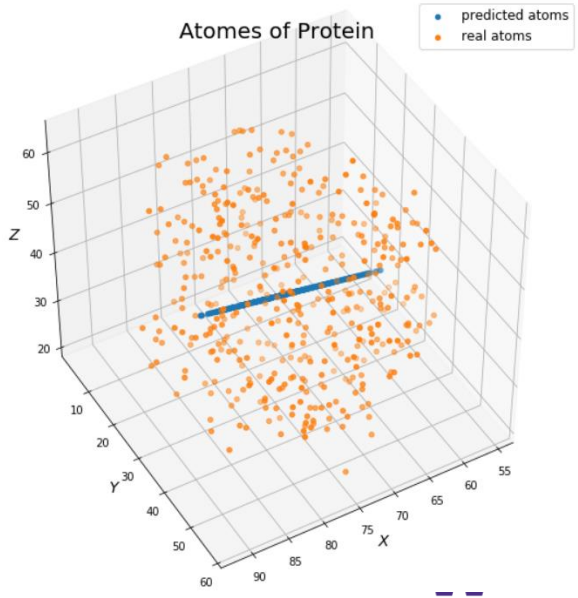
	res name	chain	ssseq
2y39	[SER, HIS, ARG, ASN, GLU, ALA, GLY, HIS]	[A, A, A, A, A, A, A, A]	[31, 32, 33, 34, 35, 36, 37, 38]
2o73	[MET, LEU, SER, ASP, ILE, GLN, THR, LYS, LEU, ...	[A, A, A, A, A, A, A, A, A, B, B, B, B, B, B, ...	[1, 167, 168, 169, 170, 171, 172, 173, 174, 1,...
3qun	[GLY, HIS, GLY, ALA, ILE, ARG, ASP, HIS, ASP, ...	[A, A, A, A, A, A, A, A, A, A, A, A, A, A, A, ...	[57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 6...
1gey	[MET, SER, THR, VAL, THR, PRO, TYR, GLN, SER, ...	[A, A, A, A, A, A, A, A, A, A, A, A, A, A, A, ...	[1, 2, 3, 4, 18, 19, 20, 21, 22, 25, 26, 27, 2...
2jfk	[MET, HIS, HIS, HIS, HIS, HIS, HIS, SER, SER, ...	[A, A, A, A, A, A, A, A, A, A, A, A, A, A, A, ...	[399, 400, 401, 402, 403, 404, 405, 406, 407, ...
3gem	[HIS, MSE, THR, LEU, SER, GLN, PRO, LYS, ASP, ...	[A, A, A, A, A, A, A, A, A, A, A, A, A, A, A, ...	[0, 1, 2, 3, 4, 183, 184, 185, 186, 187, 188, ...
3d5m	[GLU, LYS, GLY, SER, LEU, SER, ARG, ALA, ARG, ...	[A, A, A, A, A, A, A, A, A, A, A, A, A, A, A, ...	[150, 151, 152, 563, 564, 565, 566, 567, 568, ...
2z91	[SER, ALA, ALA, GLN, THR, ASN, ARG, ASP, CYS, ...	[A, A, A, A, A, A, A, A, A, A, A, B, B, C, C, ...	[128, 129, 130, 131, 132, 133, 213, 214, 215, ...
4y79	[ARG, GLY, LEU, PRO, LYS, ALA, LYS, SER, HIS, ...	[A, A, A, A, A, A, A, A, A, A, A, A, A, A, A, ...	[245, 246, 247, 248, 249, 250, 251, 252, 253, ...
3ueo	[GLY, PRO, LEU, GLY, SER, GLU, GLU, SER, LEU, ...	[A, A, A, A, A, A, A, A, A, A, A, A, A, A, A, ...	[544, 545, 546, 547, 548, 549, 550, 584, 585, ...

Neural Networks

- Application:

  1) Users can use their protein data as input and set the validation data size and seed to check if the multilayer perceptron model works well.





UNIVERSITY of WASHINGTON

Thanks for listening

UNIVERSITY *of* WASHINGTON