Preprocessing for deep learning model

Feature representation for proposed deep learning model

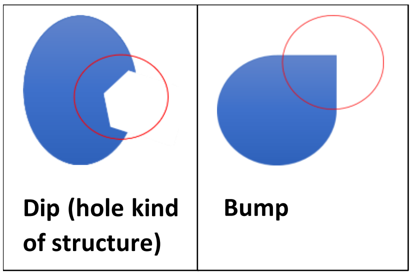

Fig.01: Keyword substructures for understand the documents

In this experiment, only surface residues' substructures are considered (without considering much about the substructure residues' atoms directions or size). These substructures can be clustered into two main kind of substructures such as Dip (hole kind of structure) or Bump which presented in Fig.1. Here the expectation of hypothesis is protein functional contribution may vary depends on the type as Dip or Bump and the residues presented. From the experiment, number of Dip structures are high, thus for this experiment only dip structures were considered (if it succeeds, I planned to apply for Bump).

These structure size may contribute the attraction of the binders (contribute to functionality) in different way. Thus, each dip structure's area(A) of outer surface, volume(V), and depth(D) are calculated. Then using those details, their channels(properties) for deep-CNN model were built as mentioned below.

Each substructures' surface residue (amino acid)'s channels were presented based on their corresponding factored biochemical properties (here the [1] biochemical properties are used). The factorization is done based on position of amino acid

center amino acid is multiplied by 1, D, 1/D, V, 1/V, A, 1/A

Group amino acid properties are multiplied by 1, V, 1/V, A, 1/A.

(The depth(D) factor is only considered in center amino acid. Overall underlying assumption is these factors may contribute the MOTIF in ONGO or TSG)

These channels(properties) were used to train the CNN-deep learning model to predict the surface-substructures-MOTIF of ONGO or TSG.

Pseudo code for feature extraction

For **PDB file** in **Functionality group PDBs**:

- Find the surface Cα atom from PDB file (this creates surface atoms coordinate file, corresponding polar-coordinate file and corresponding amino acid details for those atoms) to represent the surface residue.
- Triangulate all the surface residue (class "motif_group_intialize_pikle" is used with the help of convex hull (uses coordinate file)) triangulate all it uses polar coordinate with coordinate file.
- Find out the dip groups and bump groups first.

As proposed first dip structures were considered in initial evaluation.

- Group them with the details of the volume and surface save it with the center_amino_acid and the PDB_name detail (when it saves MOTIF contain same center atom are in one pickle file, with the ascending order of number of vertices in that MOTIF)

- Make feature set/ Channels(property) file from dip files:

  Take one pdb file's dip groups, make the pikle file of property for that group.

  The structure of on subgroup is defined in the

  Class : motif_group_pikle_to_property_pikle

  Function: property_flatten_array(self,d_v_a, center_amino_acid_property, sub_group_temp_properties)

<u>To visualize the hypothesis data is presented</u>

Input data contains property files

Sub_group_dip_1

Sub_group_dip_2

:

Sub_group_dip_i

If we take "Sub_group_dip_i" as an example, it contains the

D: Depth, V: Volume, A: Area

Center amino acid property (1 x 16)

Surrounded amino acids in the group other than center atom property (20 x16)

- (Occurrence of the amino acid (that amino acid in the vertices once) in that group counts 1
- if it occurs n times the property is multiplied by n
- If it does not occur, then multiplied the corresponding property by 0

Deep learning architecture:

Simple two-layer neural network is used as shown in the Fig.02. First layer is made like feature enhancement.

The first layer is made up with substructure Depth(D), Volume(V), Area(A), and

Properties of

center amino acid is multiplied by the fraction of 1, D, 1/D, V, 1/V, A, 1/A

Group amino acid properties are multiplied by the fraction of 1, V, 1/V, A, 1/A

(underlying assumption these factors may contribute the MOTIF in ONGO or TSG, and the Depth is only considered in center amino acid)

But this part is done in the

Class : motif_group_pikle_to_property_pikle

Function: property_flatten_array(self,d_v_a, center_amino_acid_property, sub_group_temp_properties)

And the following results only used for rest of Neural network implementation (Because these factors change one to another substructure).

In the second layer only 64 is chosen with nonlinear function Relu (assume at least 4 amino acids needed to form a substructure)

4 amino acids with 16 properties

4 x 16 = 64

weight array1: 1715 x 64

followed by relu, to make nonlinearity

weight array2:

64 x 1 to find out ONGO or TSG

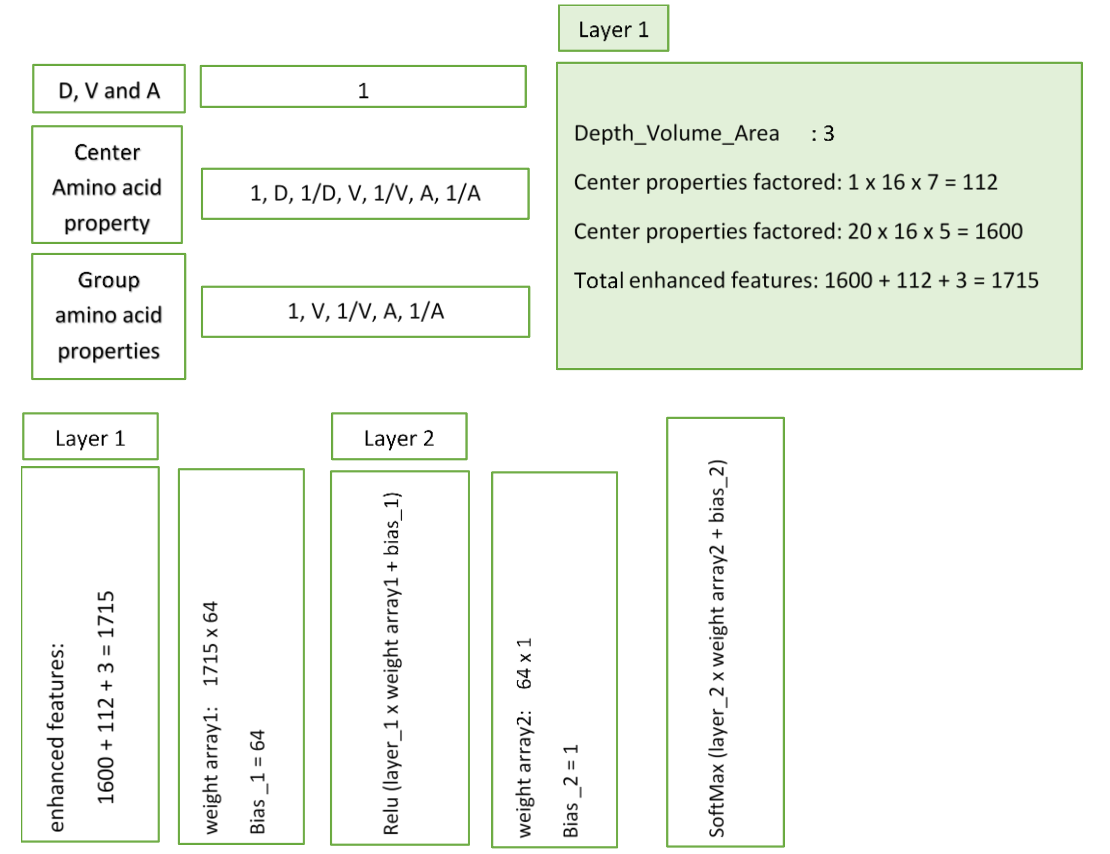
Layer 1
D, V and A
1
Center Amino acid property
1, D, 1/D, V, 1/V, A, 1/A
Group amino acid properties
1, V, 1/V, A, 1/A
Depth_Volume_Area : 3
Center properties factored: 1 x 16 x 7 = 112
Center properties factored: 20 x 16 x 5 = 1600
Total enhanced features: 1600 + 112 + 3 = 1715
Layer 1
enhanced features: 1600 + 112 + 3 = 1715
weight array1: 1715 x 64
Bias _1 = 64
Layer 2
Relu (layer_1 x weight array1 + bias_1)
weight array2: 64 x 1
Bias _2 = 1
SoftMax (layer_2 x weight array2 + bias_2)

Fig.02: Architecture of Neural network

Models’ Results and Discussion

In the training set, it has used all sub_groups(which is already randomized), and their labels arranged correspondingly. Since all pdb_sets each has at least 100 sub structures (after randomized every substructure is mixed), batch_size used as 500 in the training and go through 4000 iterations(in order to train the weights for substructure at least 5 time). Loss is calculated based on subgroup misclassification with the given label. Accuracy is calculated as how many substructures are correctly classified.

For our purpose, accuracy is no needed to be high, and losses also no needed to continuously droppable because the objective is to identify the MOTIF (useful substructures, since in the training set contains the most of substructures doesn’t belong to MOTIF). But the predictions by the model, must be labeled as 1 or 0 (1 mean motif of ONGO), but the predictions are always staying at 1. That means the network has not studied (trained properly) any think.

For this problem there are many possibilities

Mixed all substructures for training (it gives very noisy data for training because it contains substructures does not belong to MOTIF)

The way of properties represented

Lack of the architecture design

Properties represented by only vertices, but the edge information is lost, maybe add the properties of the edges and place in the amino acid columns may improve the performance. But, the main architecture design problem,

Here all substructures are mixed (all PDB substructures are mixed) and that may cause the network to learn unwanted structures (forcefully make the architecture learn noisy data), can lead to the problem.

Solution: Each PDB group substructure is not to be separated and, then over all vote of the substructure classification decide which class it is.

Eg: If PDB_id_i has “n” substructures

PDB_id_i_sub_structures	**Prediction_of_class_substructure**
PDB_id_i_sub_struc_1	0
PDB_id_i_sub_struc_2	1
PDB_id_i_sub_struc_3	1
:	:
.	.
PDB_id_i_sub_struc_n-1	0
PDB_id_i_sub_struc_n	1

The vote of the class
sum(Prediction_of_class_substructure)

Assign the highest vote class as class, and then check the label, if the label is different only increase the loss by how many predictions of substructures needed to assign as correct class. This new model may not be led for unwanted substructures misleading. After the network is trained, use the

Prediction of PDB_ids original class and substructure prediction (If both are same then that substructure may MOTIF)

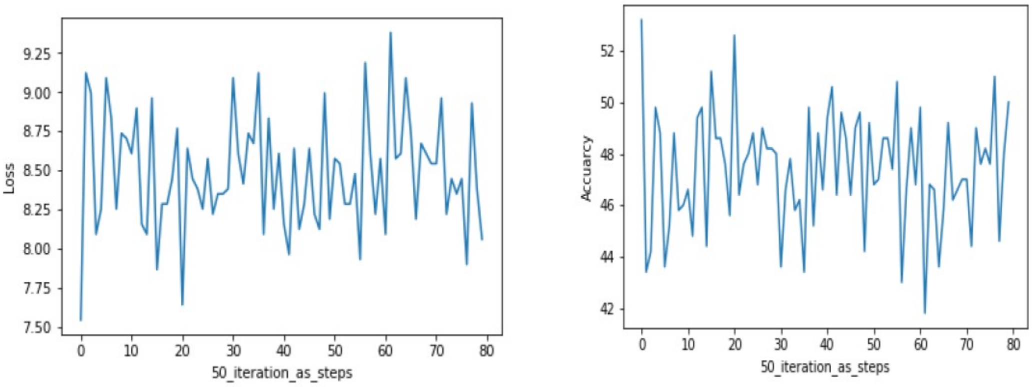

Fig.2: Accuracy and Loss of the training set

For more detailed description of the pseudo code please find the **Appendix-01**

Appendix-01: Surface motif detection deep learning approach Code files and purpose.

First, the main classes used for this coding is given

Then for each class what are main functions are used also given.

Classes

1. **motif_group_intialize_pikle:**
   This class is basically written for creating the pickle of groups of

   amino acids with neighbors' details with ascending order of length in the group
2. **motif_group_pikle_to_property_pikle:**
   This class is basically written for check the pickle of groups of amino acids PDB files

   And make the property for neural network

Important functions in class **motif_group_intialize_pikle:**

i. **distance_surface**(self, coordinates, triangle_points, missed_point_index):
   This function calculates the distance from the missed point

   coordinates: cartesian coordinates of the surface atoms

   vertices : Triangles points

   vertices_position_triangle : Which triangle from the vertices want to be considered

   missed_point_index : The point which need to calculate the distance

ii. **creating_surface_by_triangle**(self):
   This function creates the surface with triangles from

   1> create the convex hull using the library

   2> Extend the surface triangulation which should cover all surface points

   This function does

- checking_points_order: keep the missing points checking order in the **descending** order of the radius
- the triangles points are selected one by one and choose the minimum and maximum phi angle

and place with the triangle position {as HashMap triangle position to phi angles(xy_z) minimum and maximum}

For missing points in the list:

- choose the triangles to check the distance of the missed point
  to find which triangle distances, must be checked

  - first choose the triangles have theta and phi in the range of the missing coordinate

  - if that is more than one

  - then find the equation of the triangles

  - calculate the distance from the point to triangle and choose the minimum distance triangle and break that

When the above condition not satisfied

Even if that case not satisfied loose the condition even further

(When check the angles Theta and Phi angles are considered differently, from the Fig.A1.01 shown in below phi angle does not has any issues, Theta angle 180 degree and -180 degrees are same)

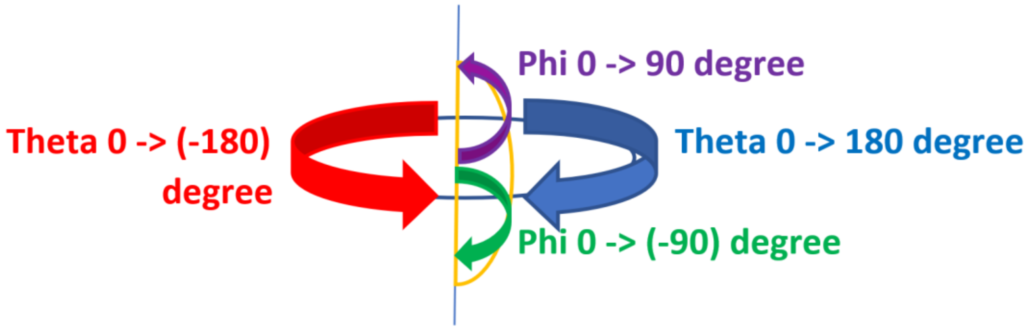

Fig.A1.01: PDB files atom in surface polar coordinates

Here only consider the 15 degree in all way's Phi +15 degree and -15 degree

Theta +15 degree and -15 degree

check which triangles middle angle fell into these ranges

25 degree = 0.2618 radians

These loosing conditions are done starting from

Then +30 -30

As mentioned for 15 degree earlier

Then +45 -45

Then +60 -60

Then +75 -75

- Consider the breaking triangle contain a, b, c as points and added point going to be d (missed point)
  then the set of new triangles are a, b, d
  b, c, d
  a, c, d

since these triangle_break and theta and phi angles are used again and again after updated also thus those are made as deep copies

- Remove the a, b, c triangle and the new triangles (a, b, d
  b, c, d

  a, c, d details (phi, theta) are added

Effectiveness of the algorithm/ visualize the algorithms' outcome is shown in the figures with Fig.A.1.2.x(x=1,2,3,4)

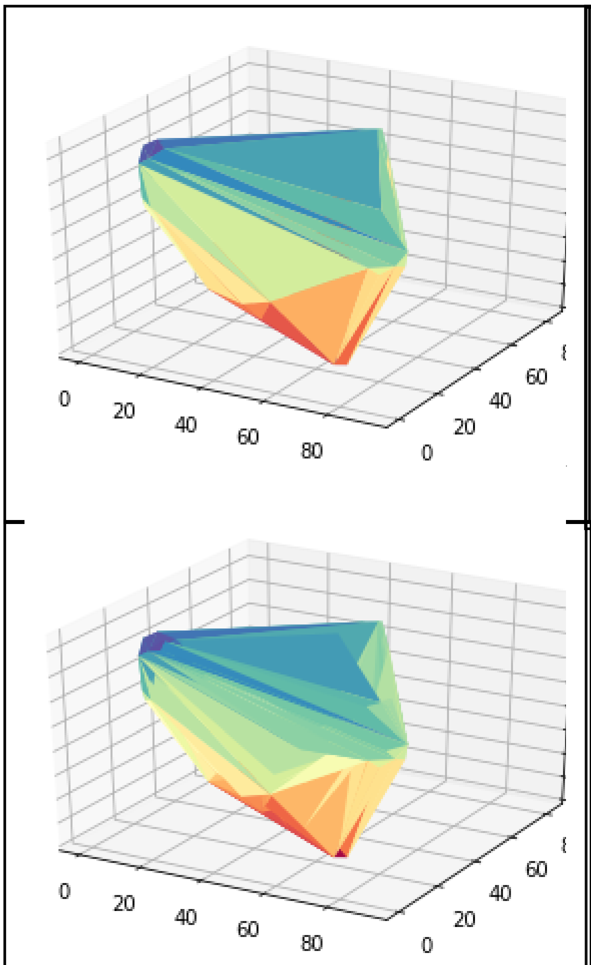

Fig.A1.2.3: 1BQU's surface visualized after the surface algorithm applied

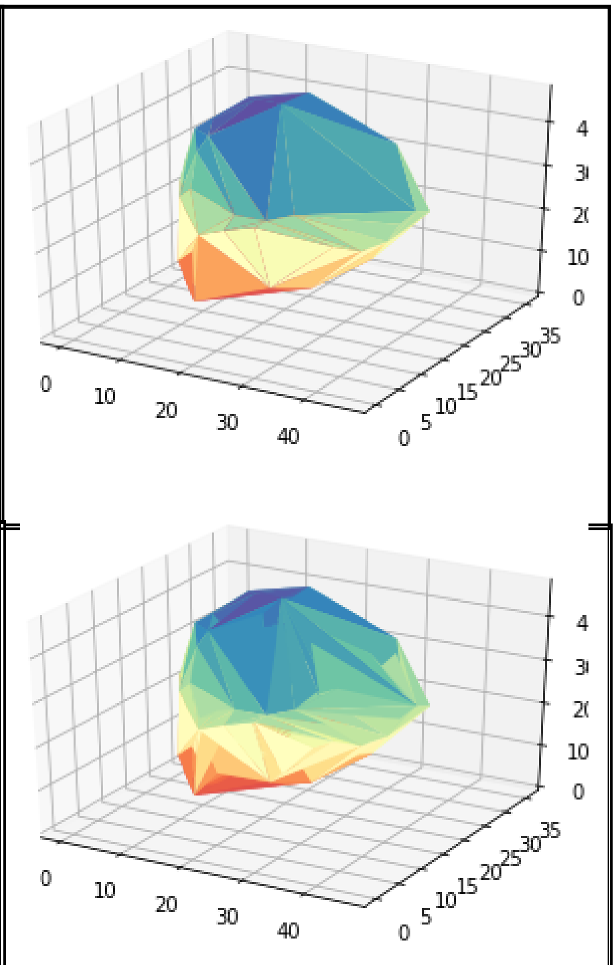

Fig.A1.2.4: 1E3G's surface visualized after the surface algorithm applied

iii. **choose_dip_bump_struct**(self): This function finds out the substructure which is dip or dump

- Calculate the normalized coordinate by divide the coordinates by resolution (it is already calculated and saved)
- To do this group_from_coordinates are used

group_from_coordinates is each surface atom is once considered as center atom(index in the group_from_coordinates mean that atom considered as center atom) and their corresponding substructures are formed from that, and these groups are represented by the index of the surface atoms.

choose the center atom once

-first find the center atom is dip or bump

where dip means the radius of center is less than all the other radius from the gravity point

bump means the radius of center is higher than all the other radius from the gravity point

For better understanding/visualization please check the hypothetical structures and Notations shown in Fig.A1.03)

these dip and dump data are saved differently to train the model differently

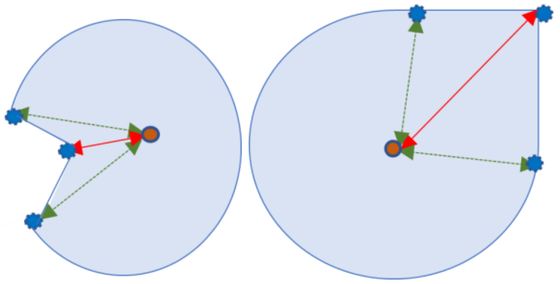

Fig.A1.03: Hypothetical representation of dip and dumb structure selection based on the center residue

Fig.A1.03's Notations dictionary	Distance from the center of gravity is annotated for the corresponding residues as
● : Center of gravity ✱ : Surface sub structure Cα residues	↔ : substructure's center residue ↔ : substructure's other residue

iv. **dip_group_amino_acid_pikle(self):**

This function includes the details of the area and volume of the substructures and other details like amino acid in the center and surrounding.

Dip case:

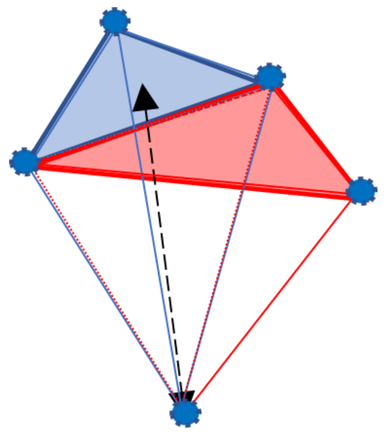

Fig.A1.04: A hypothetical structure to visualize depth, surface of the substructure and the volume formed from tetrahedrons

- surface:

  first take the surrounding atoms (other than center atom) order them in descending order with radius

  then make triangles from the order and consider previously made triangle(edges)

  Then add all the triangle surfaces.

- Volume:

  find the volume for each tetrahedron (triangle formed for surface with center atom)

  add those volumes

- Depth:

  Perpendicular distance from the first triangle found for surface

Important functions in class **motif_group_pikle_to_property_pikle:**

i. **property_flatten_array**(self,d_v_a, center_amino_acid_property, sub_group_temp_properties):

This function creates the results of the property for Neural network first layer for one group, by using other functions to retrieve the information from pickle file.

Input:

d_v_a: Depth, Volume, and Area information in order as NumPy array

center_amino_acid_property: Amino acid in the center dip's property values

sub_group_temp_properties: Amino acid properties in dip without center amino acid properties

o/p:

Combine the properties with depth, Volume and, Area

Make the property matrix combined with

area, volume, Depth

Here the center atom, and the surrounding atoms are treated in different way

The properties are attached in the order as follows

The depth, volume, area is attached

Then the Center atoms property is multiplied by

(1, depth, 1/depth, Volume, 1/Volume, 1/Area, Area)

1 x 16 x 7 = 112

Surrounding amino acid properties are make like multiplied by

(1, Volume, 1/Volume, 1/Area, Area)

20 x 16 x 5 = 1600

So total features

1600 + 112 + 3 = 1715

Less than (42 x 42 image pixel feature with one channel)

(24 x 24 image pixel feature with three channels)

MNIST: 784(28 x 28 with one channel)

Case have not considered for time constraint:

Area or

Volume or

depth

less than one is treated the same way as values higher than one

ii. **main_fn_property_from_PDB(self):**

This is main function call other helper functions to make property files as pickle

One pdb file.

First extract the general information these can be used for later programming purpose.

Initial format of the property file

Each group must represent the amino acids in the group 20 x 16

Then take each PDB ids group separately and calculate their Property values

The property of the values is represented by 20 x 16 which tells surrounded amino acid properties added.

Then depth, volume, area information is gathered from function and give to the first layer property calculating function(**property_flatten_array**)

Reference

[1] A. Tavanaei, N. Anandanadarajah, and R. Loganantharaj, “A Deep Learning Model for Predicting Tumor Suppressor Genes and Oncogenes from PDB Structure,” *bioRxiv*, p. 177378, Jan. 2017, doi: 10.1101/177378.