VERİ BİLİMİ DERGİSİ
www.dergipark.gov.tr/veri

Analysis of PDB Files and Calculating Similarity Between Two Proteins

Emre YILDIRIM[1]* , Dr. Volkan ALTUNTAŞ[2]

[1]Bursa Technical University, Faculty of Science, Computer Engineering, Bursa
[2] Bursa Technical University, Faculty of Science, Computer Engineering, Bursa

Abstract

Examining biological structures is a critical process for many broad applications, such as protein engineering, drug design and production. sThe PDB (Protein Data Bank) computer file format is the most convenient computer file format used for the identification and identification of biological molecules such as proteins. In this format, which has a typeface structure, lines indicate a particular molecule or atom and contain technical details such as the name of the atom, the name of its residue, the chain to which it is attached, and its coordinates. Various applications and intermediaries have been developed to extract various details from these files and make sense of them. These developed tools make the study and design of molecular formations simple. Examples of these technical applications are packages such as ProDy and BioJava. However, these packages may not be easy to integrate into new applications and are often integrated into large codebases for the purpose of examining their simulation on a molecular basis. They also lack techniques for updating/changing and manipulating coordinates. The Python package pdb2sql is used, which can pull files with the PDB extension into the database and present them by making them easier to parse and manipulate. In terms of different solutions, manually writing SQL code is one of the most common solutions used to query a database. However, SQL commands and queries are challenging and confusing for those outside the field and hinder the technical use of the SQL language. Right here, pdb2sql provides end users with SQL queries as an intermediary to use simple Python techniques. However, with the technique used in the article, the positive and effective sides of SQL queries are utilized and it will also be a solution to SQL query difficulties. In addition, several types of reliable classes have been created and used in pdb2sql to rotate biological and molecular structures around their axis of origin, interfere with interface and technical information, and examine structure affinity between two proteins. As a result the Python package pdb2sql; It is a PDB extension file manipulation tool that is lightweight and versatile, easy to modify, design and integrate.

Makale Bilgisi

Başvuru:
22/09/2022
Kabul:
03/10/2022

Keywords: *pdb, proteins, pdb2sql*

PDB Dosyalarının Analizi ve İki Protein Arasındaki Benzerliği Hesaplama

Özet

Biyolojik yapıları incelemek protein mühendisliği, ilaç dizaynı ve üretimi gibi birçok geniş uygulama ağı için kritik bir işlemdir. PDB (Protein Veri Bankası) bilgisayar dosya formatı, proteinler gibi biyolojik moleküllerin tanımlandırılması, kimliklendirilmesi için kullanılmakta

* Iletişim e-posta: emreyildirimn@gmail.com

olan en elverişli ve popüler bir bilgisayar dosya uzantısıdır/formatıdır. Yazı tabanı yapısında olan bu formatta, satırlar belirli bir molekülü veya atomu belirtmektedir ve atomun ismi, kalıntısının ismi, bağlı olduğu zinciri ve koordinatları gibi teknik detaylarını içerir. Bu dosyalardan çeşitli detaylar ayıklayarak anlamlandırabilmek için çeşitli uygulamalar ve aracılar geliştirilmiştir. Geliştirilen bu araçlar, moleküler oluşumların incelenmesini ve dizaynını zor olmaktan çıkartır. Bu teknik uygulamalara örnek olarak ProDy ve BioJava gibi paketler gösterilebilir. Fakat bahsi geçen paketlerin yeni uygulamalara entegre olması kolay olmayabilir ve genellikle moleküler bazda simüle edilmelerinin incelenmesi hedefinde büyük kod tabanlarına entegredir. Ayrıca koordinatları güncelleştirme/değiştirme ve manipüle etme tekniklerinden yoksundurlar. PDB uzantılı dosyaları veritabanına çekebilen ve ayrıştırma işlemi, manipüle etme gibi kullanımları kolaylaştırarak sunan Python paketi pdb2sql kullanılmıştır. Farklı çözümler açısından bakılırsa, manuel olarak SQL kodu yazmak bir veritabanına sorgu çekmek için kullanılan en yaygın çözümlerdendir. Fakat SQL komut ve sorguları alanın dışında kalan çalışmacılar için zorlayıcı ve kafa karıştırıcı olup, SQL dilinin teknik olarak kullanılmasına engel teşkil etmektedir. Tam burada pdb2sql, son kullanıcılara basit Python tekniklerini kullanımı açısından aracı olarak SQL sorgularına kolaylık sağlar. Bununla birlikte makalede kullanılmış olan teknik ile SQL sorgularının olumlu ve etkili taraflarından yararlanılır ve SQL sorgu zorluklarına da çözüm olmuş olur. Buna ek olarak, biyolojik ve moleküler yapıları orijin ekseni etrafında döndürmek, arayüz ve teknik bilgilerine müdahale etmek ve iki protein arasındaki yapı benzeşmesini incelemek için pdb2sql içerisinde birkaç tipte güvenilir sınıflar da oluşturulmuş ve kullanılmaktadır. Sonuç olarak Python paketi olan pdb2sql; hafif ve çok yönlülüğüyle birlikte, modifiye ve dizayn edilmesi ve entegrasyon işlemleri kolay bir PDB uzantılı dosya işleme aracıdır.

Anahtar Kelimeler: *pdb, proteinler, pdb2sql*

1 Introduction

V Proteins, one of the basic building blocks in living things, are in a very critical position for the continuation of life. Forming, repairing and strengthening the tissue is one of the main tasks. In addition, most events in our body occur as a result of the binding and dissolving of protein structures. For this reason, many researches are carried out on many subjects such as protein structures and similarities, functions and effects, and manipulations. The study of Nicolas Renaud and Cunliang Geng (2020) exemplifies these manipulation processes.

From the outside of today's technologies, examining the similarities and differences between the two protein structures was a very important phenomenon for the scientific world. This situation, which touches from afar or closely, from the pharmaceutical sector to all other sectors with biology content, is supported by computer science today. Considering the lack of computer scientists, it is the lack of laboratory environment and knowledge. On the other hand, the lack of those who are dealing with biology can be said to be lacking in programming languages and writing code. With the solution offered by the Protein Data Bank, which eliminates these two problems for bioinformaticians and is used as a common resource, they have the technical information of proteins and the ability to analyze proteins. The database, which offers a pdb file format, can be interpreted in different ways. In addition to manually writing Sql code, different tools have been developed to help researchers who are out of their field due to the difficulty of manual use. An example of these is the pdb2sql package, which was also used in the study. This package, available via Python, is very common due to its ease of use and easy source access.

An example of working on PDB, John L. (2006), each line represents a protein, with the content of the Pdb file format presented by the Protein Data Bank, a topic that Sussman D. Lin and O. Ritter (1998) also emphasized in their work. Many information such as protein name, inheritance, chain and coordinates can be accessed. This information, which is accessed, can also be manipulated by the researcher to serve different purposes. Researchers are right here waiting for the pdb2sql package to make their job easier. Comparing the similarities of the two proteins also becomes easier in the package, which can perform many operations such as parsing, manipulating and updating co-ordinates.

In this article, PDB sample files will be examined, the similarities between two different proteins will be calculated by examining the pdb2sql class of the

Python programming language based on SQL queries.

2 Literature Review

In the article by Nicolas R. and Cunliang G. (2020), rmsd and ligand rmds measurements of a protein model were made with the same Python package used in the article, and package usage was demonstrated. The values indicated are the average of the distance between the atoms of the interlocking proteins.

In the article by Nicolas R., Yong J., Vasant H., Cunliang G., Alexandre B., and Li C. (2020), an MPI-supported software, iScore, is presented. It is promising for modeling and studying the basic behavior and three-dimensional appearance of proteins and for other studies. The program, which automates the calculation, provides executable commands, allowing the graph of the interface to be examined. It takes advantage of the pdb2sql package in the graphing stages.

Work (2021) by Manon F. Réau , Cunliang G., Sonja G., Francesco A., Lars R., Dario F. Marzella, Nicolas R., Alexandre M., and Li C. It is on DeepRank. Three-dimensional views provide great benefits for biological analysis. Most protein interaction interfaces use deep learning technology to facilitate predictions. Data mining is also applied. The application performs classification and positioning by training on CNN, a deep learning algorithm. The difficulties here are also mentioned in the study. For both issues, the practice shows multilateralism, confronts the challenges and yields more effectively. The application made direct use of pdb2sql.

In the article by Manon F. Réau, Nicolas R, L. C. Xue, J. Bonvin (2021),

DeepRank - GNN was developed, which converts protein interfaces from pdb coordinates to graphs based on GNN architecture. The application has been created at a unitizable and customizable level. This formation was later included in the python3 package, making it user-friendly. The positive difference of the application from other graphic-based models is stated. With the Capri application, which is accepted as a success in this field, it has achieved break-even points. Compared to its predecessor DeepRank application, a significant increase in storage and speed requirements has been detected.

DeepRank-GNN is based on three-dimensional coordinate files with the pdb extension as input, and the PDB parser using SQL determines the interface between two chains using the pdb2sql class.

3 Material and Method

In the process that started with document scanning, the methods used in the literature for protein analysis and similarity measurement processes were investigated, performance and convenience criteria were prioritized, and analysis technique selection was made by making a superficial comparison. The pdb2sql technique, which is the technique used in the study of Nicolas Renaud and Cunliang Geng (2020), was used.

Python, which is an object-oriented, interactive high-level computer language, was used in the study. Thanks to Python's user-friendly libraries, it not only provides convenience in accessing information and analysis, but also offers easy and understandable use to users who are out of the field.

Protein data bank PDB (Protein Data Bank) was used to access real protein data. This data set is also used for the observation of PDB, protein-nucleid acid interactions, which provide useful data that can be used in every field from protein synthesis of proteins and nucleic acids to health and biomedicine and have wide applications. In the study by Muhammed Radifar, Nünung Yuniarti, and Enade Perdana (2013), the PyPLIF Python script was used to analyze the interaction between protein and ligand, but pdb2sql was chosen because this application would not provide much convenience for biologists and researchers outside the field of computer science.

After accessing the PDB extension protein and nucleic acid set resources, pdb2sql, a Python language package, was used to parse, manipulate and design the data it provided, to subject it to processes such as similarity measurement and calculation, and to make sense of the PDB data. More information for the pdb2sql package is available from the package documentation (https://pdb2sql.readthedocs.io/).

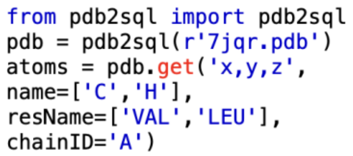
Spyder, an open source IDE written in Python and used for Python development, was used as the working environment. It is a powerful development environment with advanced editing, interactive testability, debugging and introspection capabilities. IPython and NumPy are also packages that support popular Python libraries like SciPy or matplotlib. Spyder was chosen as the development environment due to its easy-to-use and graphical support. More information about Spyder can be found in its documentation (https://docs.spyder-ide.org/current/index.html).

On the PDB files, the technical information of the determined atom was accessed by using the Python package Pdb2sql. Since anything to be manipulated needs to be seen and known first, after this process is done, the data is thrown into the numpy array for the coordinate update and design process, and the update process is performed. The identification interface class was used to find the contacting atoms of protein complexes. All atoms and residues of the interface of the protein data of interest defined by the contact distance of 10 Å were rotated. Then, using the StructureSimilarity class, the similarity measurements between the two proteins were calculated.

3.1 Accessing Technical Data in PDB Files

Each attribute can be used for filtering to select the desired atom. With the example code block in Code 1, it is provided to access the x, y and z coordinate information of those containing valine and leucine amino acid residues among carbon and hydrogen atoms.

From the Pdb2Sql library, the data of the relevant pdb file has been pulled into the pdb variable. Then, the x, y and z coordinate information of the data drawn to the variable named atoms, C and H atoms, and values connected to the A chain of valine and leucine residues were filtered.

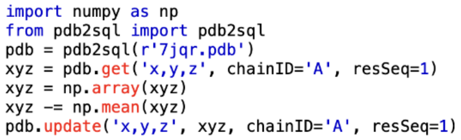


```
from pdb2sql import pdb2sql
pdb = pdb2sql(r'7jqr.pdb')
atoms = pdb.get('x,y,z',
name=['C','H'],
resName=['VAL','LEU'],
chainID='A')
```

Code 1. Extracting x,y,z coordinates of specific atoms, chains and proteins of inheritance from a pdb file

3.2 Manipulating Data in PDB Files

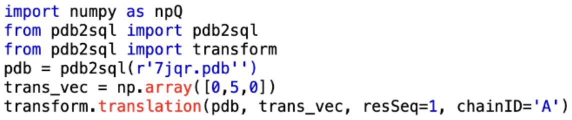
Manipulation of pdb files is available through the pdb2sql package. In the examples where updating the coordinate values will be mentioned, the variables should be assigned to the numpy array. In Code 2, the coordinates of the atoms in the first residue of the A chain are obtained and translated on the origin by the arithmetic mean of the coordinates. Then, the coordinate data in the data set was updated.



```
import numpy as np
from pdb2sql import pdb2sql
pdb = pdb2sql(r'7jqr.pdb')
xyz = pdb.get('x,y,z', chainID='A', resSeq=1)
xyz = np.array(xyz)
xyz -= np.mean(xyz)
pdb.update('x,y,z', xyz, chainID='A', resSeq=1)
```

Code 2. Manipulation of atomic data in the first residue of chain A

In Code 3, the atoms in the first residue of the A chain are rotated on the Y-axis by 5 Å (Ångström), which is one hundred millionth of 5 centimeters. This is done using the translation function of the transform module.

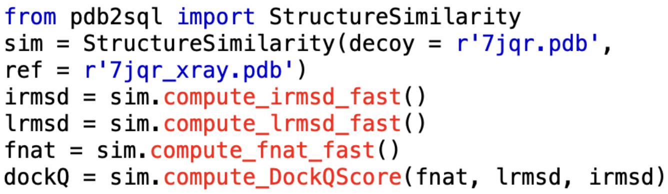


```
import numpy as npQ
from pdb2sql import pdb2sql
from pdb2sql import transform
pdb = pdb2sql(r'7jqr.pdb'')
trans_vec = np.array([0,5,0])
transform.translation(pdb, trans_vec, resSeq=1, chainID='A')
```

Code 3. 5Å unit rotation of atomic coordinates in the first residue of chain A

3.3 Using StructureSimilarity Module

The StructureSimilarity module was used to calculate the similarities between the two protein structures. Values such as rmsd, ligand rmsd and DocQ are calculated and compared to highlight the distinctive features by comparing the attributes of the proteins with each other. All the necessary methods to interlock the structures, as in the transform example, must be done before this step. Information on how to make these calculations is shared in Code 4. pdb files that have similarity information from the StructureSimilarity module; The irmsd, lrmsd and fnat values are calculated by the respective functions and the dockQ score is revealed.



```
from pdb2sql import StructureSimilarity
sim = StructureSimilarity(decoy = r'7jqr.pdb',
ref = r'7jqr_xray.pdb')
irmsd = sim.compute_irmsd_fast()
lrmsd = sim.compute_lrmsd_fast()
fnat = sim.compute_fnat_fast()
dockQ = sim.compute_DockQScore(fnat, lrmsd, irmsd)
```

Code 4. Structure Similarity Calculations

In Code 5, the required code block is specified to compare the obtained similarity results with another similarity result output. In this way, the results obtained can be directly compared with the results of reliable sources.

```
from pdb2sql import StructureSimilarity
import os
ref = '7jqr.pdb'
decoys = os.listdir('./decoys')
irmsd = {}
for d in decoys:g
sim = StructureSimilarity(d, ref)
irmsd[d] = sim.compute_irmsd_fast(method='svd', izone='7jq
```

Code 5. Comparison of different result data and pdb2sql data

4 Results

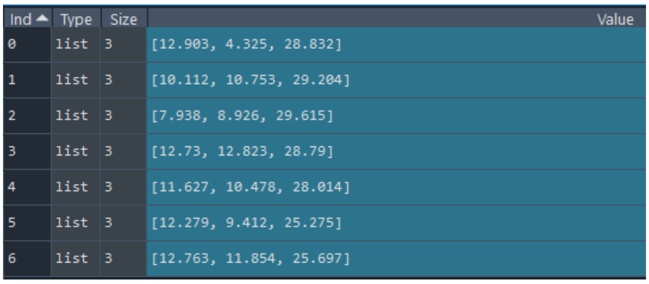
The information to be obtained from the sampled PDB data file will be as in Table 1, respectively, in SQL column logic.

Table 1. Attributes and definitions of atoms in PDB files.

Attribute	Definition
Serial	Atom Serial Number
Name	Atom Name
altLoc	Alternate Location Indicator
resName	Residue Name
chainID	Chain Identifier
resSeq	Residue Sequence Number
iCode	Code for Insertion of Residues
X	Orthogonal Coordinates for x in Angstroms
Y	Orthogonal Coordinates for y in Angstroms
Z	Orthogonal Coordinates for z in Angstroms
Occ	Occupancy
Temp	Temperature Factor
Element	Element Symbol
Model	Model Serial Number

4.1 *Extracting Data From PDB Files*

It is an important detail for the disciplines interested in biology to reveal the three-dimensional structures of biomolecules. In the formation of the three-dimensional image, the coordinate information of the relevant protein and molecule is directly used. Therefore, the coordinates of the atoms containing valine and leucine amino acid residues obtained in Figure 1 for carbon and hydrogen atoms could be easily reached. Thanks to the Spyder IDE, an explanatory image and a visual of the three-dimensional information of the coordinates can be obtained.

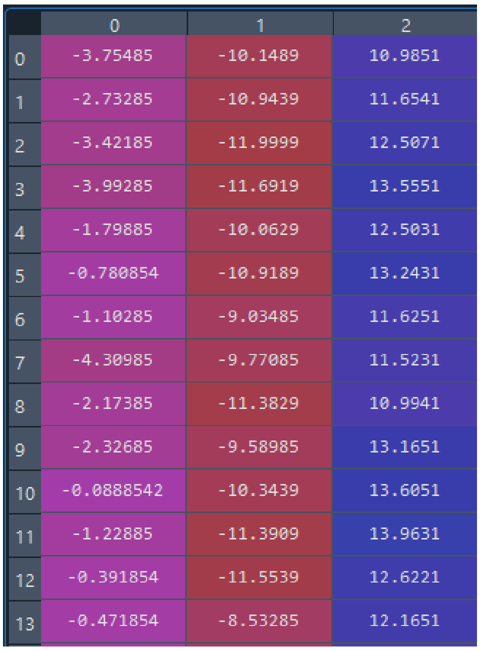


Ind	Type	Size	Value
0	list	3	[12.903, 4.325, 28.832]
1	list	3	[10.112, 10.753, 29.204]
2	list	3	[7.938, 8.926, 29.615]
3	list	3	[12.73, 12.823, 28.79]
4	list	3	[11.627, 10.478, 28.014]
5	list	3	[12.279, 9.412, 25.275]
6	list	3	[12.763, 11.854, 25.697]

Figure 1. Among the carbon and hydrogen atoms, the x, y, z coordinates of those containing valine and leucine amino acid residues

4.2 *Manupilating PDB Files*

In light of the importance of visualizing proteins and biological phenomena, understanding the relationships and interactions between structures is critical. As there are many programs dealing with this issue, operations such as manipulating three-dimensional structures can be easily provided thanks to the convenience provided by the pdb2sql package. As a result of this phenomenon that encourages active learning, advances in computation and skill acquisition in the relevant field overcomes the barrier between the challenges of configuring three-dimensional data. As seen in Figure 2, pdb2sql provides visual inspection by making it easy to manipulate coordinates. Thanks to the update and translation functions provided by the package, these operations can be easily achieved without facing difficulties.



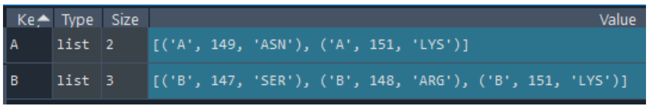
	0	1	2
0	-3.75485	-10.1489	10.9851
1	-2.73285	-10.9439	11.6541
2	-3.42185	-11.9999	12.5071
3	-3.99285	-11.6919	13.5551
4	-1.79885	-10.0629	12.5031
5	-0.780854	-10.9189	13.2431
6	-1.10285	-9.03485	11.6251
7	-4.30985	-9.77085	11.5231
8	-2.17385	-11.3829	10.9941
9	-2.32685	-9.58985	13.1651
10	-0.0888542	-10.3439	13.6051
11	-1.22885	-11.3909	13.9631
12	-0.391854	-11.5539	12.6221
13	-0.471854	-8.53285	12.1651

Figure 2. Coordinate data as a result of manipulating the atom data in the first residue of chain A

4.3 Identifiying Interface

The interface class, which is composed of the pdb2sql class, is used to detect the protein atoms and residues that come into contact with each other between two chains at a certain distance. It is used to determine and examine the interfaces of protein structures. As can be seen in Figure 3, all atoms and residues of the "7jqr.pdb" interface defined by the 10 Å contact distance have been rotated.

As can be seen in Figure 3, for the distance of 10 Å, the information about which atoms are within 10 units of each other has been reached. Therefore, since proximity to each other is an important detail in terms of similarity, easy access to this information is an undeniably positive situation.



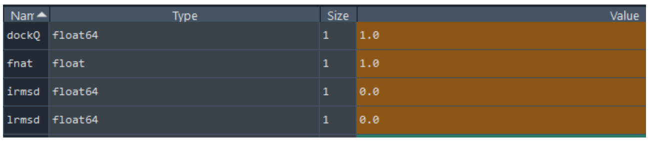
Ke	Type	Size	Value
A	list	2	[('A', 149, 'ASN'), ('A', 151, 'LYS')]
B	list	3	[('B', 147, 'SER'), ('B', 148, 'ARG'), ('B', 151, 'LYS')]

Figure 3. Data from rotation of interface atoms and residues defined by 10 Å contact distance

4.4 Structure Similarity Calculation

In the last step towards obtaining affinity and similarity criteria, similarity comparison becomes easy by obtaining dockQ, fnat, irmsd and lrmsd values of molecules. Thanks to the detection of protein-ligand interactions, it is possible to analyze the data obtained with reliable and external source programs that are accepted by the literature.

Here, the necessity of relying on outsourced programs and the necessity of comparing the results obtained with the output of another program can be seen as a negative situation. Since laboratory comparison is not possible for all researchers, accuracy should be determined by such comparisons.



Nam	Type	Size	Value
dockQ	float64	1	1.0
fnat	float	1	1.0
irmsd	float64	1	0.0
lrmsd	float64	1	0.0

Figure 4. Similarity measurement results for the StructureSimilarity class

It has been shown how to use and analyze similarity classes by analyzing our own data with programs with high experimental accuracy and widely used in the literature. Figure 5 shows the similarity between the data obtained from reliable sources and the data we obtained in this examination, which was based on the proximity of each other and the docking data of the proteins by rotating them around the origin. Based on ProFit software as an external source.

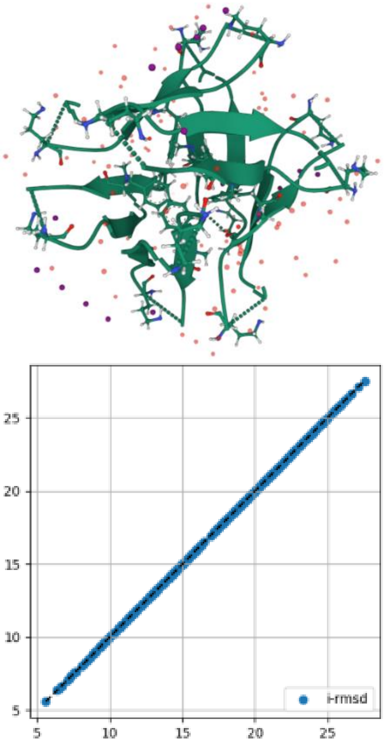


Figure 5. Top – 7 JQR Protein 3D Image Bottom – Structure Comparison of similarity data obtained with Similarity class with Profit x: Profit y : Pdb2Sql

5 Argument

As stated in the article RCSB Protein Data Bank (2021), as a result of the examination of PDB data which is very valuable for research and needs to be protected, the fast and practical package of the easy and practical Python language in which we perform operations such as parsing and manipulating PDB formatted files, data that are almost exactly close to popular similarity criteria calculation software have been obtained by using pdb2sql. The effect of the pdb2sql package, which can be learned easily even by examining the documentation in a short time, on obtaining the results, and the close results of the package and other accepted similarity criterion calculation software, proves the ease of similarity research on two proteins. As can be seen in the results, thanks to the regular and understandable attributes of the PDB format files, similarity calculations on two atoms were easily performed, and similarity criteria values between two atoms were calculated.

The pdb2sql package, which can be considered as the beginning of many studies such as coordinate updating and drawing technical data, provides convenience on the application and provides a versatile and positive use. However, the need for an external source for similarity measurements will also pose a problem for researchers who cannot reach the data that can be obtained through laboratory research. The need to rely on an external source places pdb2sql in a position that is not yet self-sufficient and cannot prevent it from being an externally dependent approval mechanism. This negative situation can also be evaluated positively in terms of having the feature of being compared with other sources with an optimistic attitude. It is thought that the pdb2sql package, which is fast, practical and easy to learn, will be more functional and close to professional use in the future, as it is open to be strengthened by developers.

Acknowledgement

The authors would like to thank TÜBİTAK ULAKBİM, High Performance and Grid Computing Center (TRUBA), Bursa Technical University High Performance Computing Laboratory.

References

[1] Muhammad Radifar, Nunung Yuniarti, Enade Perdana Istyastono, PyPLIF: Python- based Protein-Ligand Interaction Fingerprinting, (2013).

[2] Nicolas Renaud, Yong Jung, Vasant Honavar, Cunliang Geng, Alexandre M.J.J. Bonvin, Li C. Xue , iScore: An MPI supported software for ranking protein–protein docking, (2020).

[3] Documentation of Pdb2sql

[4] Nicolas Renaud, Cunliang Geng, The pdb2sql Python Package: Parsing, Manipulation and Analysis of PDB Files (2020).

[5] M. Réau, Nicolas Renaud, C. Xue, J. Bonvin, DeepRank-GNN: A Graph Neural Network Framework to Learn Patterns in Protein-Protein Interfaces (2021).

[6] Nicolas Renaud, Cunliang Geng, Sonja Georgievska, Francesco Ambrosetti, Lars Ridder, Dario F., DeepRank: a deep learning framework for data mining 3D protein-protein interfaces (2021).

[7] Yanyan Lan, Liang Pang, Jiafeng Guo, Jun Xu, Jingfang Xu, Xueqi Cheng, DeepRank: A New Deep Architecture for Relevance Ranking in Information Retrieval (2017)

[8] Ming Chen, Xiuze Zhou, DeepRank: Learning to rank with neural networks for recommendation (2020).

[9] Raaiha Humayun Kabir, Bisma Pervaiz, Tayyeba Muhammad Khan, Adnan Ul-Hasan, Raheel Nawaz & Faisal Shafait, DeepRank: Adapting Neural Tensor Networks for Ranking the Recommendations (2019).

[10] J. L. Sussman, D. Lin, J. Jiang, N. O. Manning, J. Prilusky, O. Ritter and E. E. Abola, Protein Data Bank (PDB): Database of Three-Dimensional Structural Information of Biological Macromolecules (1998).

[11] Helen Berman, Kim Henrick, Haruki Nakamura, John L. Markley, The worldwide Protein Data Bank (wwPDB): ensuring a single, uniform archive of PDB data (2006).

[12] Stephen K. Burley, Helen M. Berman, GerardJ.Kleywegt, John L. Markley, Haruki Nakamura & Sameer Velankar, Protein Data Bank (PDB): The Single Global Macromolecular Structure Archive (2017).

[13] Stephen K. Burley, Charmi Bhikadiya, Chunxiao Bi, RCSB Protein Data Bank: Celebrating 50 years of the PDB with new tools for understanding and visualizing biological macromolecules in 3D (2021).