User Guide for the PDB File Parser Program

Introduction into what the program is and what it does

This program is a Python program that parses PDB files provided by the user and allows the user to select what distinct functionalities they want to carry out on the PDB files they provide the program with. There are 4 different functionalities in the program, these are structural information of the protein, C-alpha distance, x,y,z coordinates for a specified amino acid and generating plots for B-factor distribution and predominant amino acid residues. All these functions will be explained in more detail in this user guide.

What is the purpose of each function ?

Function 1 - Structural information

The function structural information provides useful information relating to the number of ATOM entries and HETATM entries in a PDB file. It also outputs what unique chains the protein structure consists of. So a protein structure could have 2 chains A and B and this will be outputted to the user so that they can access this information easily for any protein structure where that information is contained in the PDB file format.

Function 2 - C-alpha distance

The function C-alpha distance calculates and outputs the distance between two alpha carbon atoms of two residues in a protein. C-alpha distances are useful as they can be used in constructing a distance matrix. A distance matrix can be used in predicting structure and dynamics of a protein.

Function 3 - B-factor distribution and Predominant Amino Acid plots

This function is for plotting B-factor distribution and predominant amino acids for protein structures in the Protein Data Bank. The B-factor distribution plots B-factor distribution, therefore allowing the user to visualise the distribution of B-factors for ATOM and HETATM records.

The Predominant Amino Acid Residues plot enables the user to visualise a bar chart on what amino acids are most prevalent in the structure of a particular protein. This is in

relation to the number of ATOM entries for that amino acid that are present in the PDB file and thus the atoms which are part of an amino acid residue in the protein.

Function 4 - X, Y and Z coordinates for a specified amino acid and visualisation on PyMOL

This function takes a PDB file from the user and asks the user which amino acid they want to extract atomic coordinates x,y,z for. The output is shown to the user and using PyMOL the output PDB file generated by the program can be visualised.

Python Installation

As the program is written in Python, Python installation is required to run the program and make use of it. Below is a link to the official Python page and this will guide you on how to install Python for the Linux operating system.

https://www.python.org/downloads/

Important note: Please install the latest version of Python

What Python packages and libraries need to be installed to use the program ?

The program uses quite a few Python modules, the Python modules used by the program will not need to installed separately. However these are the libraries and packages that will need to be installed separately:

- Biopython
- BioPandas
- Numpy
- Matplotlib

How to install these

Each of the four libraries and packages mentioned above need to be installed separately and below are links to pages where they can be installed for the Linux operating system.

Biopython

https://biopython.org/wiki/Download

BioPandas - Make sure that when you open the link below you read the section titled Requirements as there software's and packages that are required and need to be installed to use BioPandas, they will be listed and links to install those software's and packages will be provided.
http://rasbt.github.io/biopandas/installation/

Numpy
https://numpy.org/devdocs/user/install.html

Matplotlib
https://matplotlib.org/users/installing.html

Starting the program

To start the program open the Linux terminal.

Go to the directory that you have put the python script PDB_File_Parser.py in, this is the python script for this program. Once you are in that directory, in the terminal type:
python3 PDB_File_Parser.py
This command will launch this python script and start the program. It is important to note that if you are in a directory and that directory contains other directories and in one of those directories is your Python script then make sure you use the cd command to move into the directory where your python script is in otherwise you cannot launch your Python script. Therefore it is a good idea and definitely recommended that your Python script and all of the PDB files that you want to use in the program are within the same directory.

It is important that you do this but if you forget to put your PDB files in the same directory then don't worry, in the user guide their will be an explanation on how to change your current working directory in the program.

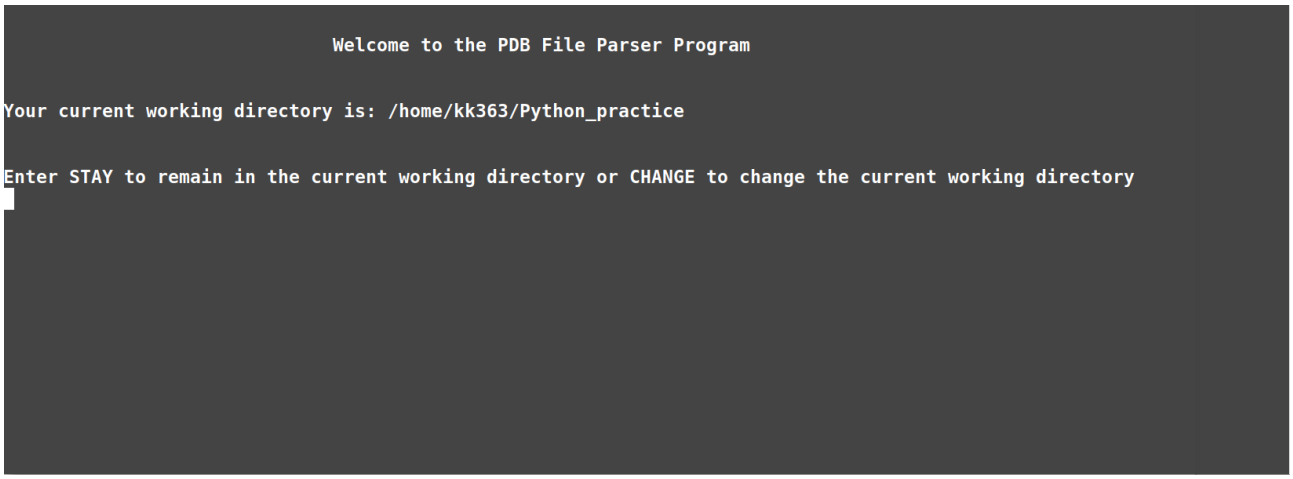
Step by step instructions on using the program and navigating your way through it

After you have read and gone through all the instructions in this section, run the program using a PDB file. This will act as a test file and let you play around with the programs functions and gain a better understanding of how the program works.

Below is a link to the RCSB PDB: Homepage and the structure 3EIY where you can download this structure in PDB format and use it as your test file for this program.
https://www.rcsb.org/structure/3EIY

Step 1 - Remaining in or changing your current working directory

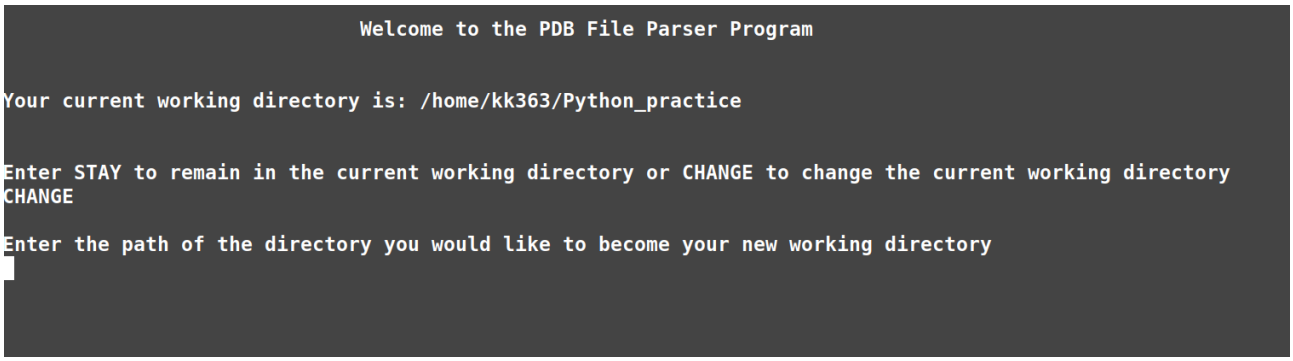
After you have typed the command python3 PDB_File_Parser.py the program will launch and you will be presented with the welcome screen as shown below.



This is the welcome screen of the program and is what you will initially see in the program. Your current working directory which is where you have run the Python script from will be displayed and obviously will be different from the one displayed in the figure above. The program will prompt you to enter STAY to remain in the current working directory, or to enter CHANGE to change the current working directory. In the terminal type either STAY or CHANGE based on what you want to do. Make sure to enter STAY or CHANGE in capital letters as shown in the program otherwise the program will ask you for input again.

What if you want to change your current working directory and you enter CHANGE ?

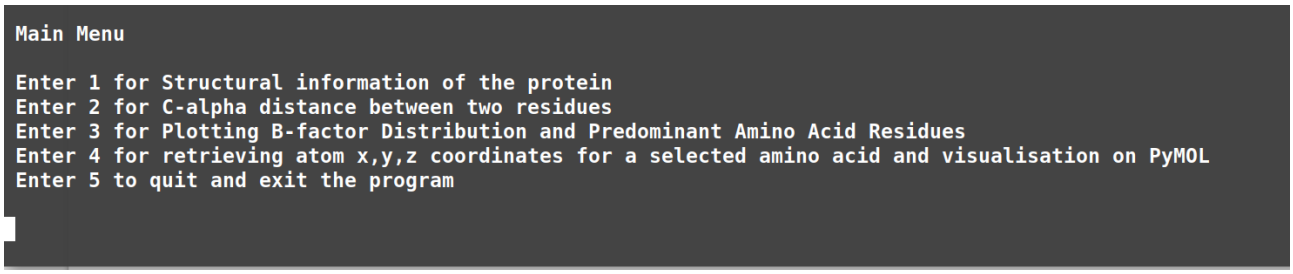
If once you start the program you realise all the PDB files you would like to work with are within another directory, the program gives you the option to change your current working directory as displayed in the figure below. To change your current working directory enter the path of the directory you would like to become your new working directory. For e.g. if you would like your new working directory to be Program_PDB you would modify the path in the figure below from /home/kk363/Python_practice to /home/kk63/Program_PDB



It is important to note that if you provide the program with a path that does not exist the program will tell you that the path does not exist and will ask you to enter a path that does exist.

Step 2 – Main Menu and selecting a function

Once Step 1 is complete, the next stage of the program is to select a function based on what you would like to do to the PDB files you want to use in the program. As displayed in the figure below, entering 1 will select function 1 and so on. An option is also provided to exit and quit the program, to do this you need to enter 5 in the terminal.



Step 3 – How to navigate your way through the functions

In function 1, 2 and 4 you will be prompted by the program to enter the filename of the PDB file you would like to work with. Function 3 is different in this regard to function 1,2 and 4 and this will be explained in this user guide.

Important note: If you enter an invalid filename 5 times, the next time you enter a filename, even if it is valid you will be redirected to the stay or change directory part of the program.

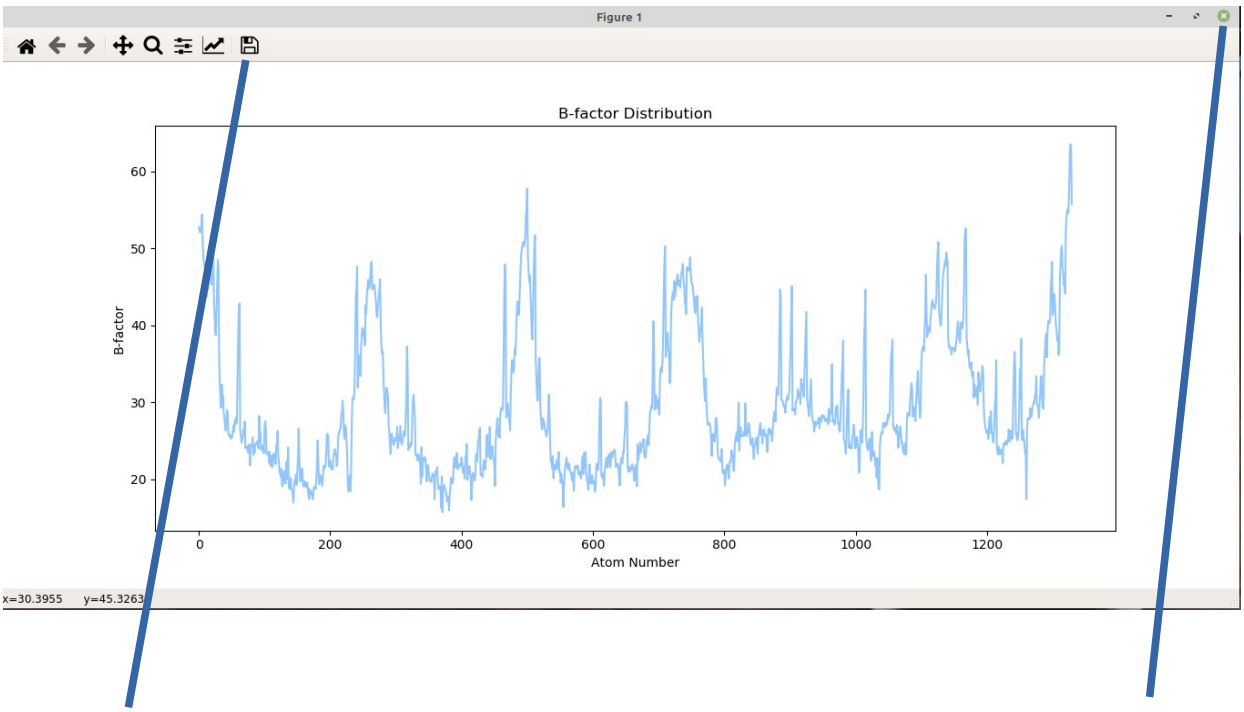
How is function 3 different and how to operate it

In function 3 the PDB structure you enter when asked by the program to enter any PDB structure from the Protein Data Bank, you do not have to have downloaded and saved that PDB file, you can enter any PDB structure from the Protein Data Bank. Also it is important to follow what the program prompts you to do, enter a valid PDB entry and don't put the extension .pdb . If you do include the .pdb extension, the program will stop running therefore you will have to run the Python script PDB_File_Parser.py again using the command python3 PDB_File_Parser.py as mentioned before. This will launch the program again and then you can continue to use the program.

How to manipulate, work with and save the plots generated in function 3

The annotated figure below will explain how to manipulate, work with and save the plots the program generates.

This is an example of a B-factor Distribution plot generated for 3EIY



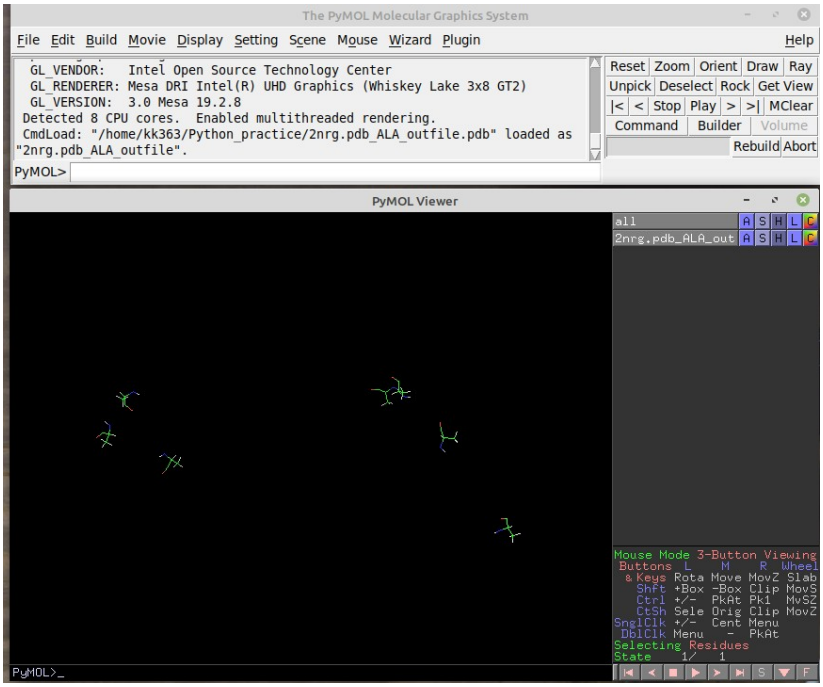
Click this icon to save the plot. The other icons can be selected to manipulate and do different things with the plot.

Click this icon to exit and return to the program.

Function 4 and visualising the output file/files in PyMOL

Once PyMOL is launched by the program you can visualise the output file/files that are generated. The program will have told you what name/names the output file/files were saved under. The file/files will be saved in the current working directory that you are using in the program.

Below is a figure with guidance on how to navigate your way through PyMOL to find and open the output file/files you want to visualise.



Click on File and in File click on Open and select the file that is the output file the program has generated for you, this will display an image which you can visualise in PyMOL.

An example of this is the figure on the left, the figure generated is a 3D visualisation of alanine residues in the structure 2NRG.

To find out more information about PyMOL and the features available in PyMOL use the link below.

https://pymol.org/2/