

PepVis – Peptide Virtual Screening Pipeline

Operating Manual

by

*Mr.A.Samdani & Dr.V.Umashankar
Centre for Bioinformatics
Vision Research Foundation
Sankara Nethrlaya
Tamil Nadu, India
Chennai 600006*

Website: https://www.visionresearchfoundation.org/department_of_bioinformatics/department_of_bioinformatics.html

Email : yumashankar@gmail.com

Email: samdani1593@gmail.com

Distributed under GNU/GPL license V3.

PREREQUISITES TO RUN PepVis:

PepVis GUI run in python2.7

Linux Operating system 64 bit with bash shell

Peptide Modelling and optimization:

GNU parallel (/usr/local/bin)

Link to download and installation of GNU Parallel (<http://ftp.gnu.org/gnu/parallel/>)

Modpep

Link to download the standalone version of Modpep-v1.0 (<http://huanglab.phys.hust.edu.cn/software.php>).

Gromacs

Link to download Gromacs (<http://www.gromacs.org/Downloads>).

MGL TOOLS

Ensure to set path to the path where mgltools have been installed (export MGLROOT=/usr/local/MGLtools-1.5.7)

Link to download and installation of MGLTOOLS (<http://mgltools.scripps.edu/downloads>)

Virtual screening:

Autodock Vina (<http://vina.scripps.edu/download.html>)

- ✓ vina (/usr/local/bin)
- ✓ vina_split (/usr/local/bin)

ZDock-3.0.2 (<http://zdock.umassmed.edu/software/>)

ZRANK2 (<http://zdock.umassmed.edu/software/>)

ROSETTA (<https://www.rosettacommons.org/software/license-and-download>)

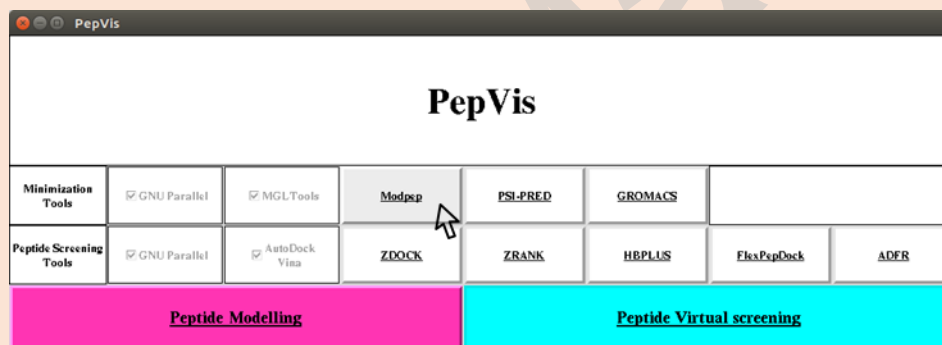
AutoDock CrankPep [ADCP] (<https://ccsb.scripps.edu/adcp/>)

Software configuration:

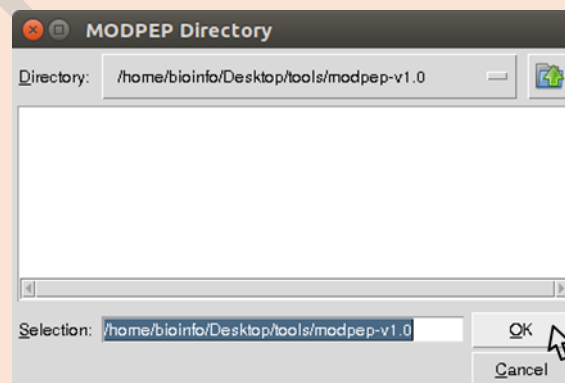
1. On execution of Pepvis, The list of software's configured properly will appear along with a as tick in the checkbox. If you want to configure the tools. For example: ModPep



2. Click the Modpep button



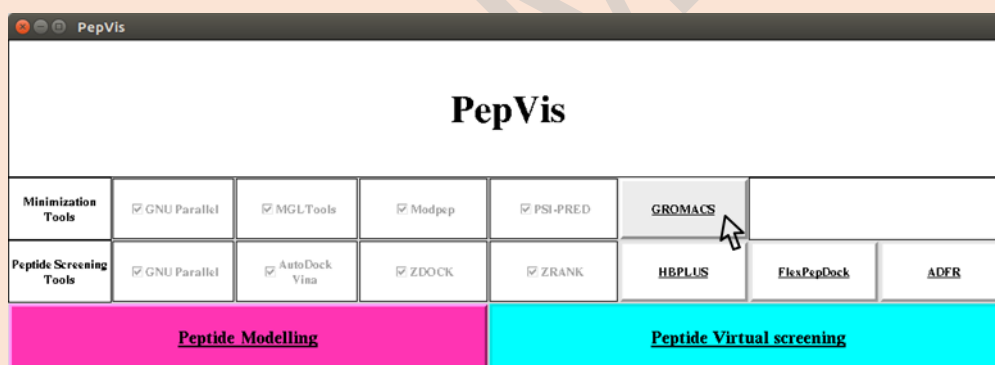
3. Change to the respective directory of where modpep executable was extracted and click ok.



4. If the correct path is provided, the button will be changed to checkbox with tick.



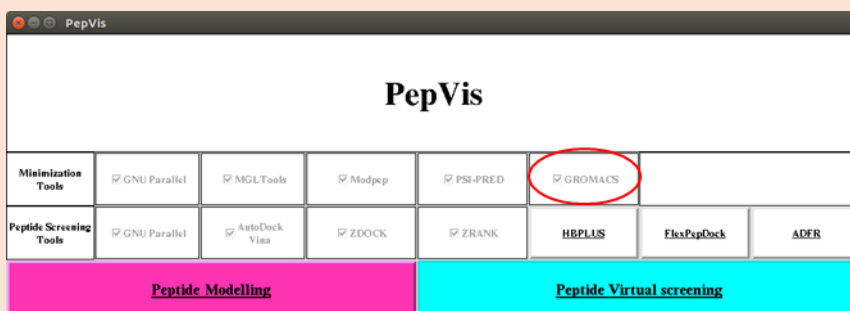
5. Same procedure can be followed for configuring other tools also. Ex: in the case of gromacs, click the gromacs button.



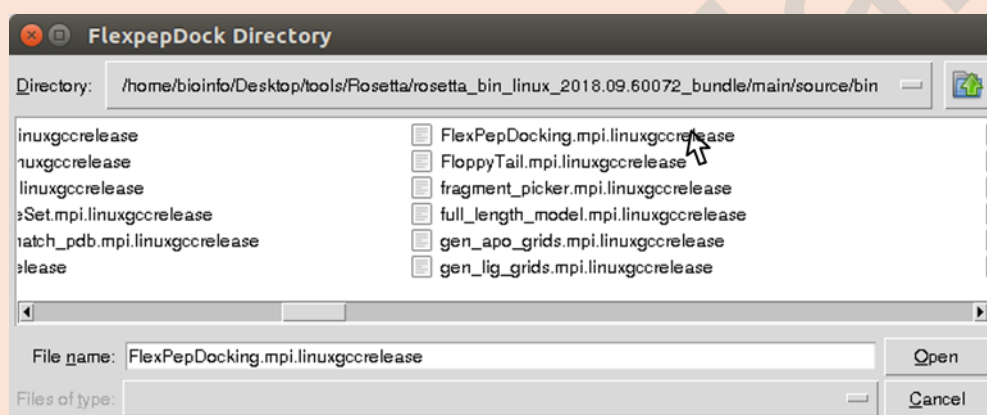
6. A pop-up window will be prompted for obtaining the gromacs prefix like gmx_mpi. Enter the appropriate prefix and click enter button.



7. When the correct gromacs prefix is given the button will change into checkbox with tick.



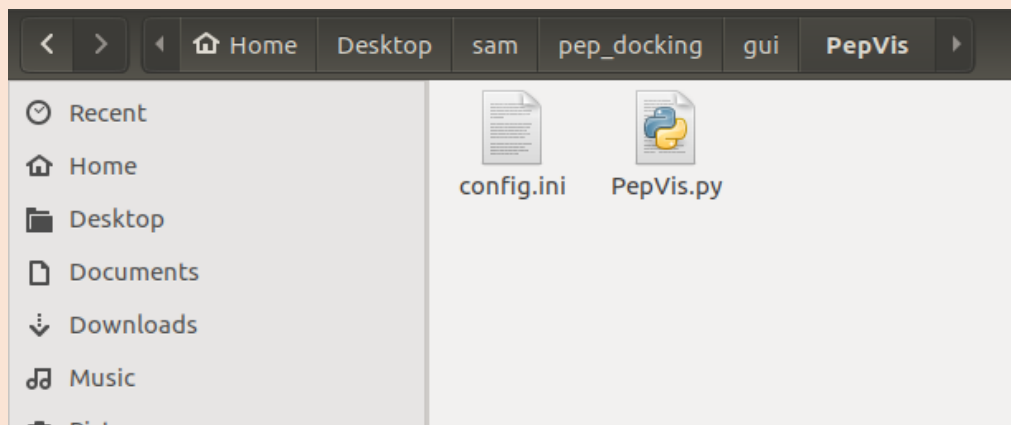
8. In the case of FlexPepDock, go in to the main/source/bin and select the respective FlexPepDocking.mpi.linuxgccrelease exe for configuration.



9. The same procedure can be followed for the configuration of other tools also.

10. config.ini file will be created in the current directory which contains all the path information for the tools which will be used when starting the script next onwards without need of configuring each and every time and you can check the check box ticked for confirmation also.





Config.ini

```
Open ▾  Save
```

```
1 [softwares]
2 MODPEP = /home/bioinfo/Desktop/tools/modpep-v1.0
3 ZDOCK = /home/bioinfo/Desktop/sam/other/pep_combo/zdock3.0.2_linux_x64
4 PSIPRED = /home/bioinfo/Desktop/tools/psipred
5 ZRANK = /home/bioinfo/Desktop/sam/other/pep_combo/zrank2_linux
6 GROMACS_PREFIX = gmx_mpi
7 HBPLUS = /home/bioinfo/Desktop/tools/hbplus
8 FLEXPEPDOCK = /home/bioinfo/Desktop/tools/Rosetta/rosetta_bin_linux_2018.09.60072_bundle/main/source/bin/FlexPepDocking.mpi.linuxgccrelease
9 ADFRRROOT = /home/bioinfo/ADFRsuite-1.0
10
```

OPERATING INSTRUCTIONS:

Peptide Preparation:

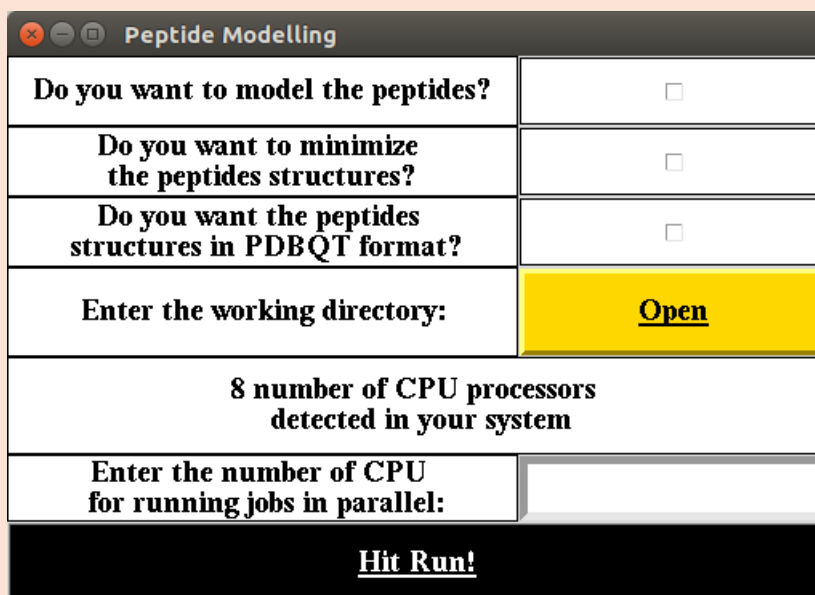
1. After checking the proper installation of softwares, Run the python script PepVis.py in the terminal



2. Select the Peptide Modelling button



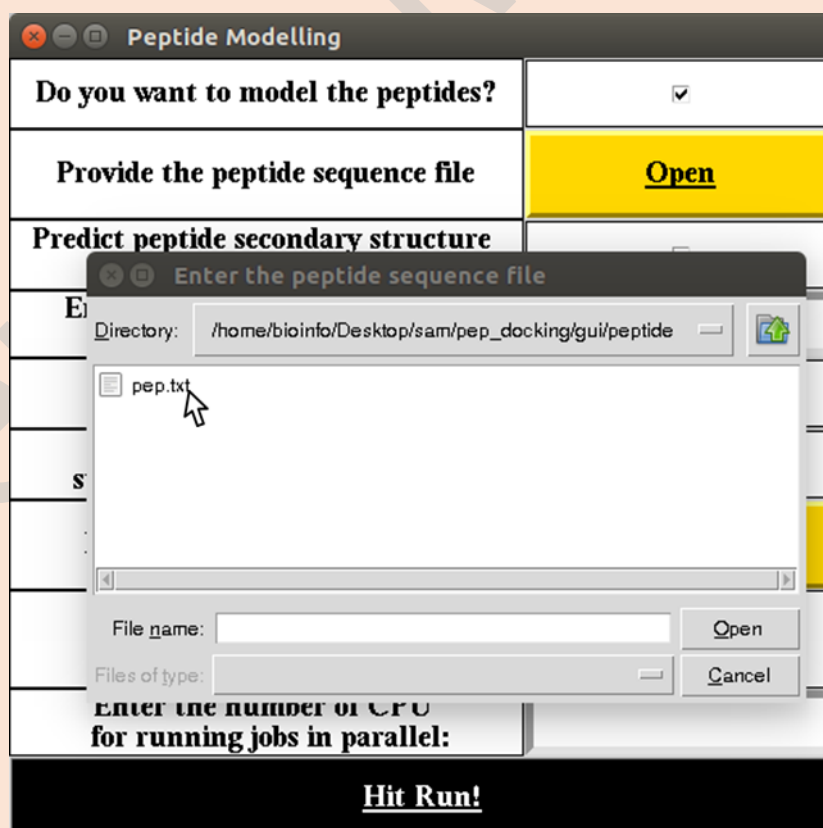
3. Pop-Up window will prompt from getting input for the peptide structure preparation and optimization.



The screenshot shows the 'Peptide Modelling' window with the following options:

| | |
|---|--------------------------|
| Do you want to model the peptides? | <input type="checkbox"/> |
| Do you want to minimize the peptides structures? | <input type="checkbox"/> |
| Do you want the peptides structures in PDBQT format? | <input type="checkbox"/> |
| Enter the working directory: | Open |
| 8 number of CPU processors detected in your system | |
| Enter the number of CPU for running jobs in parallel: | |
| Hit Run! | |

4. Provide the required option in the inputs area:



The screenshot shows the 'Peptide Modelling' window with the following options:

| | |
|---|-------------------------------------|
| Do you want to model the peptides? | <input checked="" type="checkbox"/> |
| Provide the peptide sequence file | Open |
| Predict peptide secondary structure | <input type="checkbox"/> |
| Enter the peptide sequence file | |
| Enter the number of CPU for running jobs in parallel: | |
| Hit Run! | |

The 'Enter the peptide sequence file' dialog is open, showing the directory: /home/bioinfo/Desktop/sam/pep_docking/gui/peptide. The file 'pep.txt' is selected. The dialog also includes fields for 'File name:' and 'Files of type:', and buttons for 'Open' and 'Cancel'.

5. Provide the required input option of steps to perform the peptide modelling and optimization in the input prompt.

| Peptide Modelling | |
|---|-------------------------------------|
| Do you want to model the peptides? | <input checked="" type="checkbox"/> |
| Provide the peptide sequence file | Open |
| Predict peptide secondary structure using PSI-PRED? | <input type="checkbox"/> |
| Enter the number of models to generate per peptide | 1 |
| Do you want to minimize the peptides structures? | <input checked="" type="checkbox"/> |
| Provide the peptide structure directory | Open |
| Do you want the peptides structures in PDBQT format? | <input checked="" type="checkbox"/> |
| Do you want to the inactivate peptide torsions? | <input type="checkbox"/> |
| Enter the working directory: | Open |
| 8 number of CPU processors detected in your system | |
| Enter the number of CPU for running jobs in parallel: | |
| Hit Run! | |

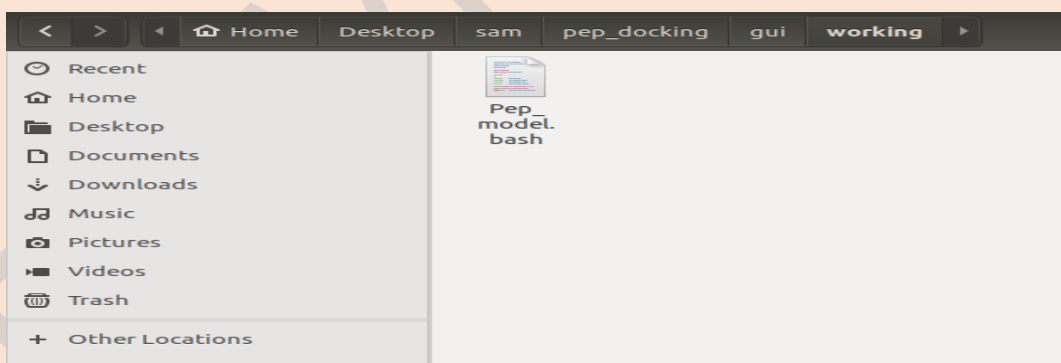
6. Provide the working directory path details

| Peptide Modelling | |
|--|-------------------------------------|
| Do you want to model the peptides? | <input checked="" type="checkbox"/> |
| Provide the peptide sequence file | Open |
| Predict peptide secondary structure using PSI-PRED? | <input type="checkbox"/> |
| Enter the number of models to | 1 |
| Enter the working directory | |
| Directory: /home/bioinfo/Desktop/sam/pep_docking/gui/working | |
| Selection: /home/bioinfo/Desktop/sam/pep_docking/gui/working | |
| OK Cancel | |
| 8 number of CPU processors detected in your system | |
| Enter the number of CPU for running jobs in parallel: | |
| Hit Run! | |

7. Enter the number of CPU for running in parallel and hit run button.

| Peptide Modelling | |
|---|-------------------------------------|
| Do you want to model the peptides? | <input checked="" type="checkbox"/> |
| Provide the peptide sequence file | Open |
| Predict peptide secondary structure using PSI-PRED? | <input type="checkbox"/> |
| Enter the number of models to generate per peptide | 1 |
| Do you want to minimize the peptides structures? | <input checked="" type="checkbox"/> |
| Provide the peptide structure directory | Open |
| Do you want the peptides structures in PDBQT format? | <input checked="" type="checkbox"/> |
| Do you want to the inactivate peptide torsions? | <input type="checkbox"/> |
| Enter the working directory: | Open |
| 8 number of CPU processors detected in your system | |
| Enter the number of CPU for running jobs in parallel: | 8 |
| 8 Jobs will run in parallel | |
| Hit Run! | |

8. Now a bash script will be generated in the working directory provided. The bash script can be run in terminal as bash PepMod.bash alone. The script will run all the provided input steps and peptide datasets will be stored in the working directory.



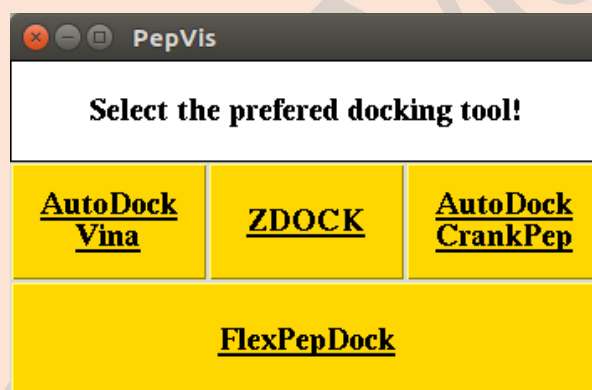
Peptide Virtual Screening:

A: Using AutoDock Vina:

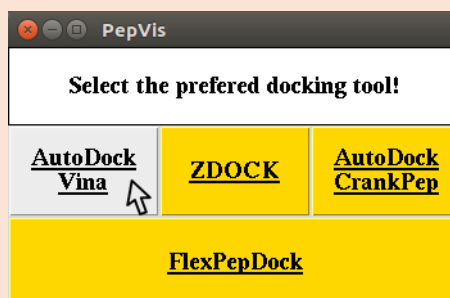
1. Start the Virtual screening script by executing the command `python PepVis.py` in the terminal the main window will pop-up.



2. The pop-up window with different docking tools or flexible refinement alone buttons will be displayed.



3. Select the required button (AutoDock Vina)



4. Input window for performing AutoDock Vina will pop-up.

| AutoDock Vina: Inputs | |
|---|--------------------------|
| Enter the peptide structures in pdbqt format: | <u>Open</u> |
| Enter the receptor structures: | <u>Open</u> |
| Enter the working directory: | <u>Open</u> |
| 8 number of CPU processors detected in your system | |
| Enter the No. of CPU for single vina job: | 8 |
| 1 Jobs will run in parallel | |
| Enter the exhaustiveness: | 8 |
| Do you want to rescore using ZRANK? | <input type="checkbox"/> |
| Do you want to run Flexible refinement using FlexPepDock? | <input type="checkbox"/> |
| <u>Hit Run!</u> | |

5. Provide the directory containing prepared peptide structures in pdbqt format.

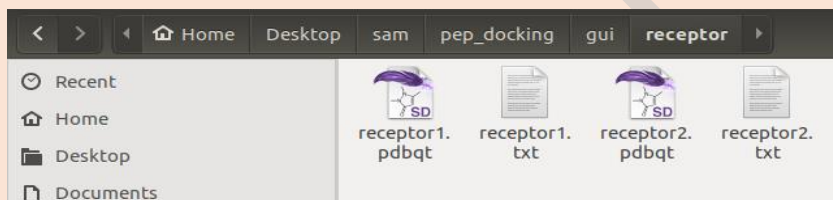
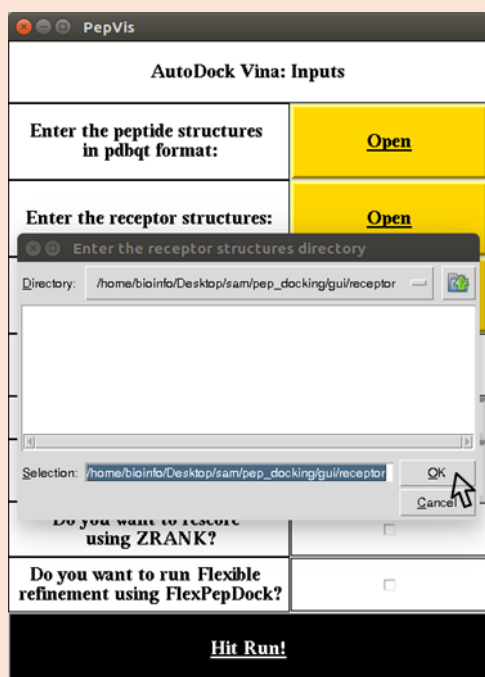
Enter the peptide structures directory

Directory: /home/bioinfo/Desktop/sam/pep_docking/gui/peptide

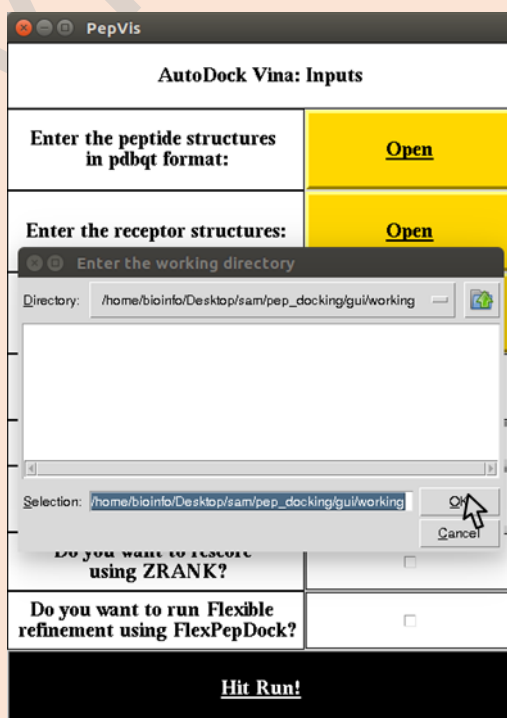
Selection: /home/bioinfo/Desktop/sam/pep_docking/gui/peptide

OK Cancel

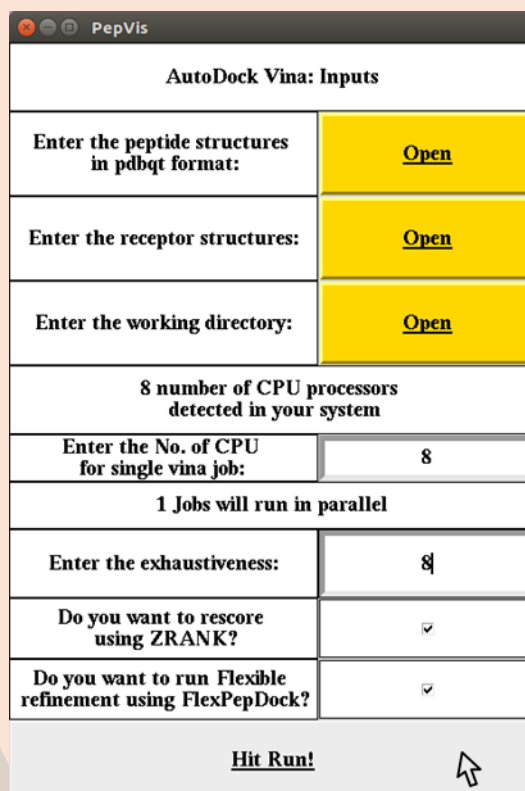
6. Provide the directory containing the receptor structures and respective configuration files (both receptor and config file should be same). The configuration file should contain the defined active site region to be docked.



7. Provide the path to working directory.

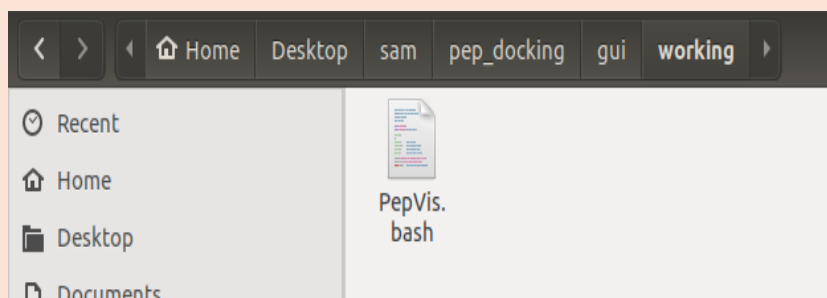


8. Provide the number of CPU for running single vina job and the exhaustiveness value. Can check the box if rescoring with ZRANK needed to be performed and also flexible refine the complex structures. Hit Run button after providing all the required inputs.



| AutoDock Vina: Inputs | |
|---|-------------------------------------|
| Enter the peptide structures in pdbqt format: | Open |
| Enter the receptor structures: | Open |
| Enter the working directory: | Open |
| 8 number of CPU processors detected in your system | |
| Enter the No. of CPU for single vina job: | 8 |
| 1 Jobs will run in parallel | |
| Enter the exhaustiveness: | 8 |
| Do you want to rescore using ZRANK? | <input checked="" type="checkbox"/> |
| Do you want to run Flexible refinement using FlexPepDock? | <input checked="" type="checkbox"/> |
| Hit Run! | |

9. The PepVis.bash script will be generated in the working directory based on the input provided. The bash script can be run in the terminal by typing bash PepVis.bash from the working directory which will perform functions of the provided input and the results will be stored in the working directory.

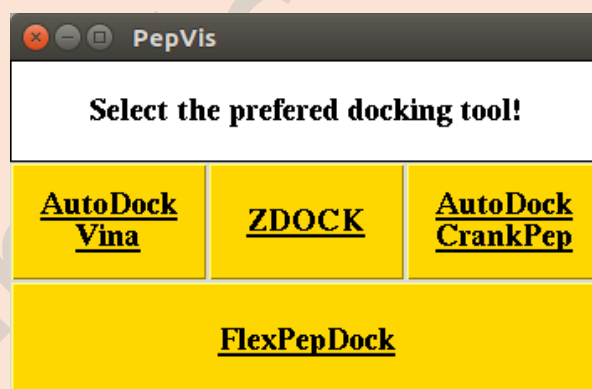


B: Using ZDOCK:

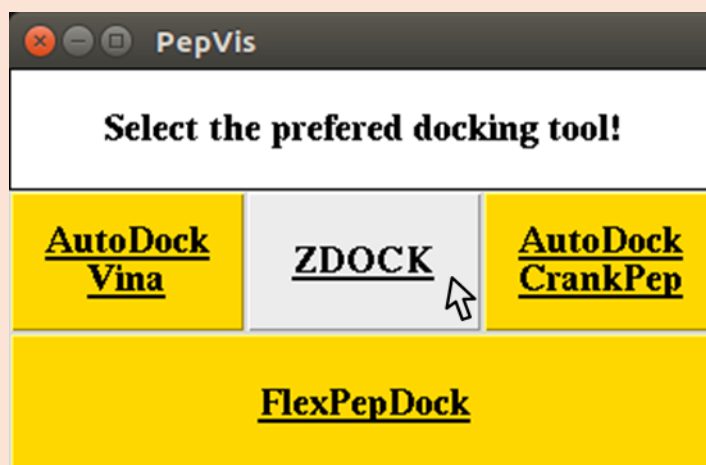
1. Start the script by executing the command `python PepVis.py` in the terminal and the main window will pop-up.



2. The pop-up window with different docking tools or flexible refinement alone buttons will be displayed.



3. Select the ZDOCK button.

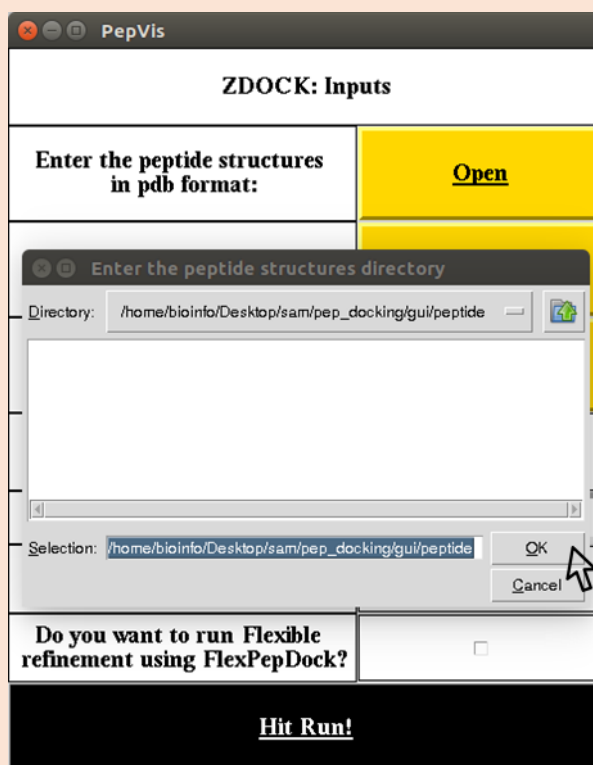


4. The pop-up window for obtaining the ZDOCK input will be prompted.

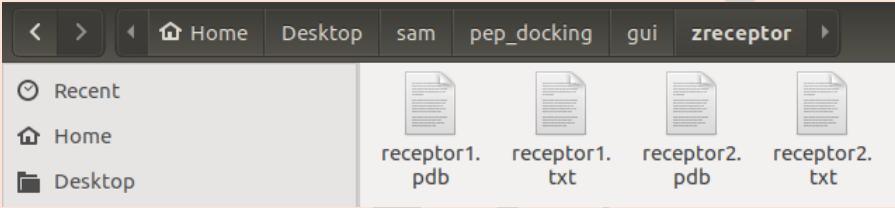
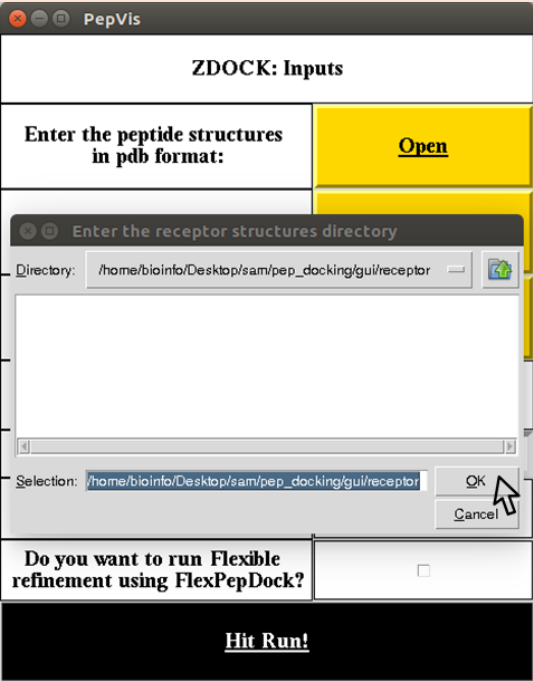
The screenshot shows a window titled 'PepVis' with a dialog box titled 'ZDOCK: Inputs'. The dialog contains several input fields and buttons. The first three rows are for entering peptide structures, receptor structures, and the working directory, each with an 'Open' button. The fourth row shows '8 number of CPU processors detected in your system'. The fifth row is for entering the number of CPU for running jobs in parallel, with a text box containing '8'. The sixth row shows '8 Jobs will run in parallel'. The seventh row is for rescore using ZRANK, with a checkbox. The eighth row is for flexible refinement using FlexPepDock, with a checkbox. The bottom row is a black button labeled 'Hit Run!'.

| ZDOCK: Inputs | |
|---|--------------------------|
| Enter the peptide structures in pdb format: | Open |
| Enter the receptor structures: | Open |
| Enter the working directory: | Open |
| 8 number of CPU processors detected in your system | |
| Enter the number of CPU for running jobs in parallel : | 8 |
| 8 Jobs will run in parallel | |
| Do you want to rescore using ZRANK? | <input type="checkbox"/> |
| Do you want to run Flexible refinement using FlexPepDock? | <input type="checkbox"/> |
| Hit Run! | |

5. Provide the directory containing prepared peptide structures in .pdb format

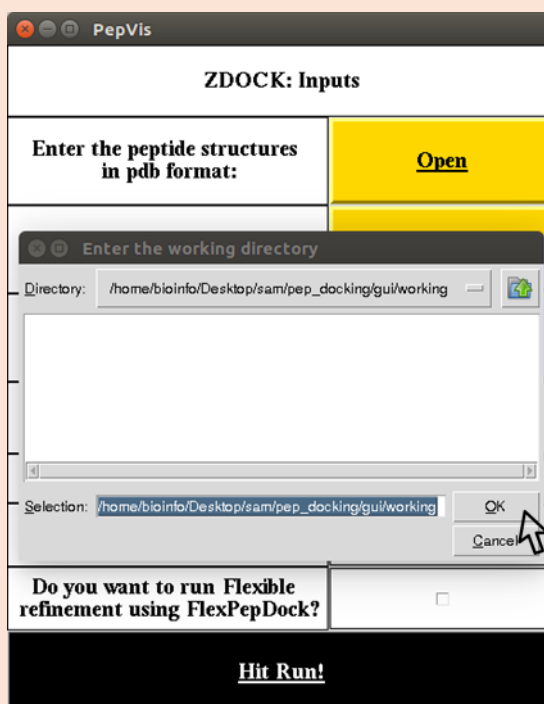


6. Provide the directory containing receptor and active site txt file containing the active site residues. The active site can be specified as residue number and chain as example shown below.



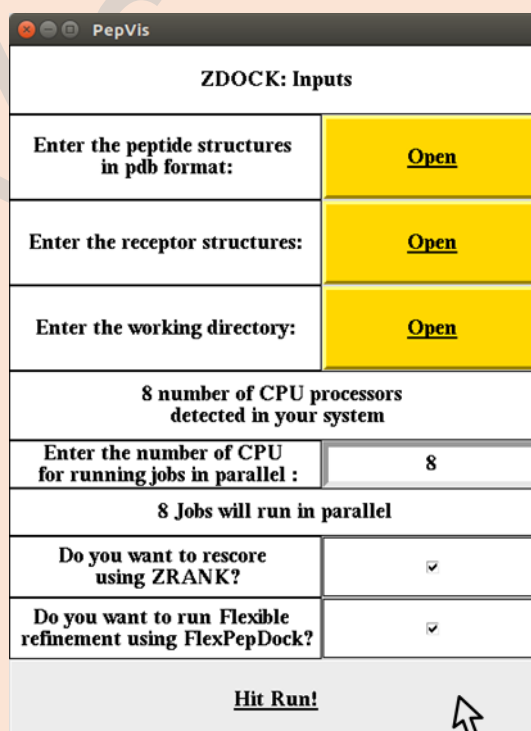
| Open ▾ | | |
|--------|-----|---|
| 1 | 45 | A |
| 2 | 46 | A |
| 3 | 47 | A |
| 4 | 48 | A |
| 5 | 49 | A |
| 6 | 50 | A |
| 7 | 51 | A |
| 8 | 52 | A |
| 9 | 53 | A |
| 10 | 54 | A |
| 11 | 56 | A |
| 12 | 74 | A |
| 13 | 75 | A |
| 14 | 76 | A |
| 15 | 78 | A |
| 16 | 79 | A |
| 17 | 112 | A |
| 18 | 113 | A |

7. Provide the path to working directory.



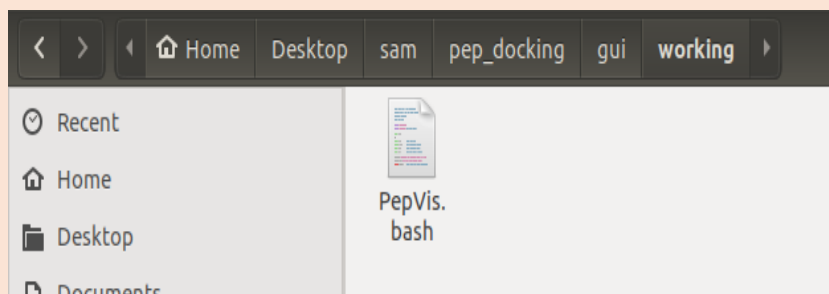
The screenshot shows the 'PepVis' application window with the 'ZDOCK: Inputs' section. The 'Enter the peptide structures in pdb format:' field has a yellow 'Open' button. The 'Enter the working directory:' field has a text input showing '/home/bioinfo/Desktop/sam/pep_docking/gui/working' and a yellow 'Open' button. A dialog box titled 'Enter the working directory' is open, showing the same path in the 'Directory:' field and a 'Selection:' field. The 'OK' button is highlighted with a mouse cursor. Below the dialog box, the 'Do you want to run Flexible refinement using FlexPepDock?' checkbox is unchecked. At the bottom, there is a 'Hit Run!' button.

8. Provide the number of CPU's to run in parallel. Check the button if Rescore using ZRANK or not? If the Flexible refinement of the complex need to be performed?



The screenshot shows the 'PepVis' application window with the 'ZDOCK: Inputs' section. The 'Enter the peptide structures in pdb format:' field has a yellow 'Open' button. The 'Enter the receptor structures:' field has a yellow 'Open' button. The 'Enter the working directory:' field has a yellow 'Open' button. Below these, it says '8 number of CPU processors detected in your system'. The 'Enter the number of CPU for running jobs in parallel :' field has a value of '8'. Below this, it says '8 Jobs will run in parallel'. The 'Do you want to rescore using ZRANK?' checkbox is checked. The 'Do you want to run Flexible refinement using FlexPepDock?' checkbox is checked. At the bottom, there is a 'Hit Run!' button with a mouse cursor pointing to it.

9. The PepVis.bash script will be generated in the working directory based on the input provided. The bash script can be run in the terminal by typing `bash PepVis.bash` from the working directory which will perform functions of the provided input and the results will be stored in the working directory.

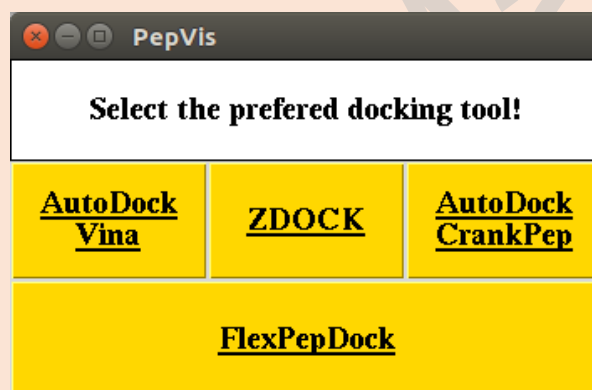


C: Using ADCP:

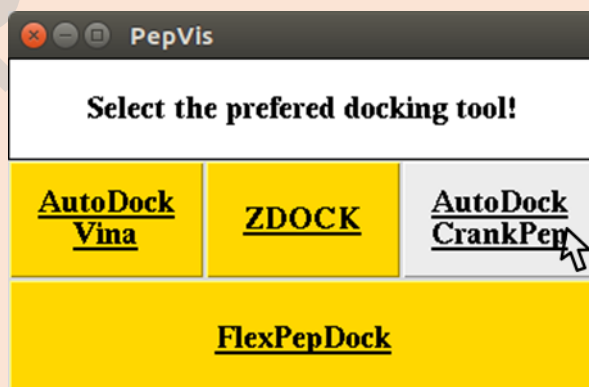
1. Start the script by executing the command `python PepVis.py` in the terminal and the main window will pop-up.



2. The pop-up window with different docking tools or flexible refinement alone buttons will be displayed.



3. Select the AutoDock CrankPep button.



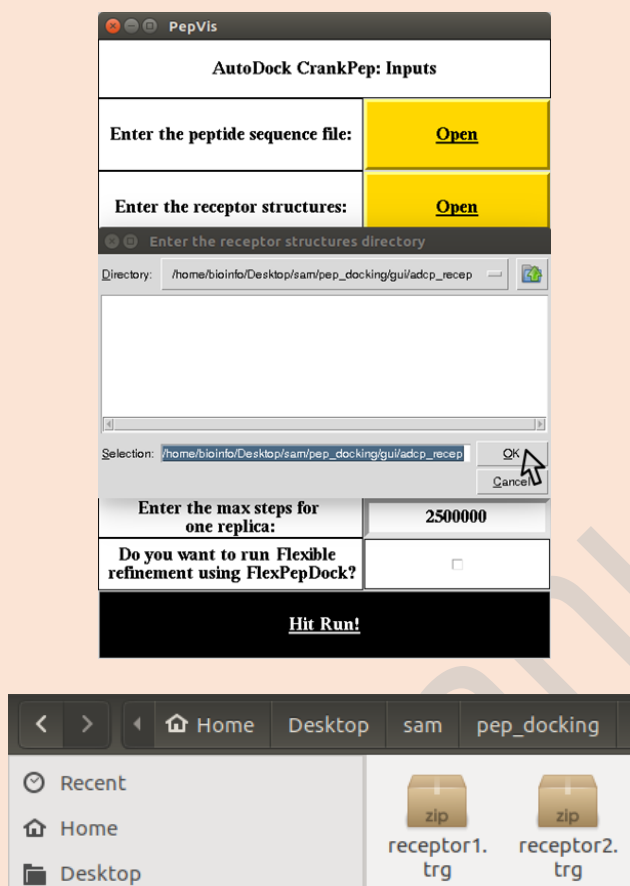
4. The new pop-up window for providing the inputs for AutoDock CrankPep will be prompted

| AutoDock CrankPep: Inputs | |
|---|--------------------------|
| Enter the peptide sequence file: | Open |
| Enter the receptor structures: | Open |
| Enter the working directory: | Open |
| 8 number of CPU processors detected in your system | |
| Enter the number of CPU for running jobs in parallel : | 8 |
| 8 Jobs will run in parallel | |
| Enter the number of replicas: | 50 |
| Enter the max steps for one replica: | 2500000 |
| Do you want to run Flexible refinement using FlexPepDock? | <input type="checkbox"/> |
| Hit Run! | |

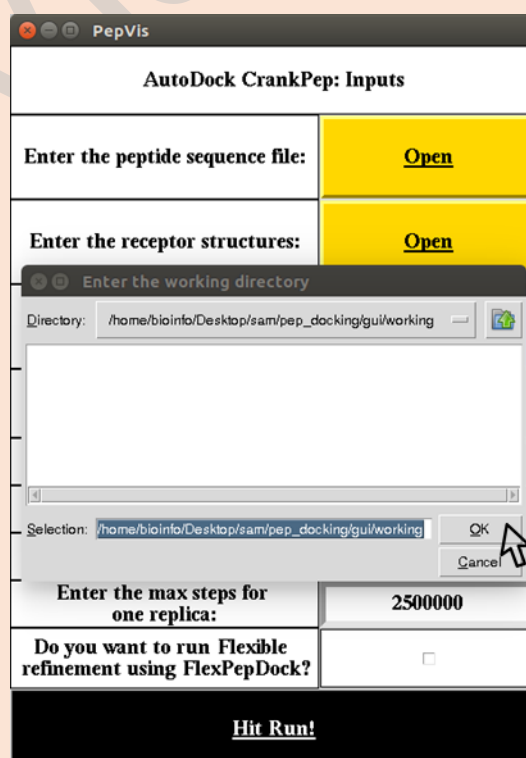
5. Provide the peptide sequences containing text file.

| AutoDock CrankPep: Inputs | |
|---|--------------------------|
| Enter the peptide sequence file: | Open |
| Enter the receptor structures: | Open |
| 8 number of CPU processors detected in your system | |
| Enter the number of CPU for running jobs in parallel : | 8 |
| 8 Jobs will run in parallel | |
| Enter the number of replicas: | 50 |
| Enter the max steps for one replica: | 2500000 |
| Do you want to run Flexible refinement using FlexPepDock? | <input type="checkbox"/> |
| Hit Run! | |

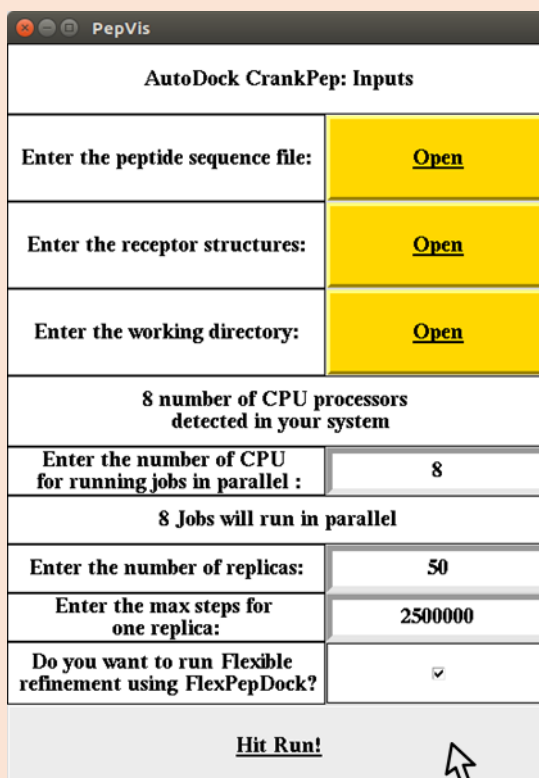
6. Provide the directory containing the receptor in .trg file format prepared using AGFR.



7. Provide the working directory.

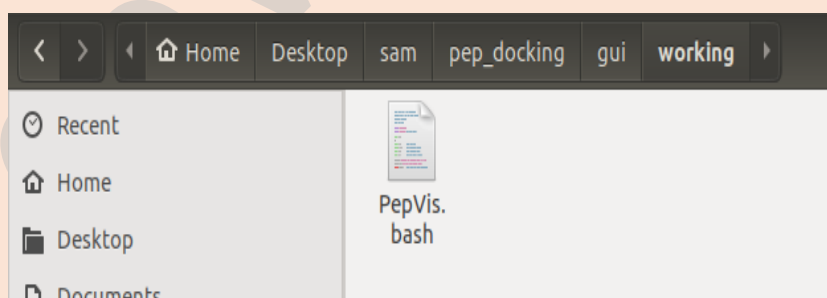


8. Provide other inputs such as Number of CPU for parallel run, number of replicas to run, maximum steps for each replica and check the box if the complex needed to be flexible refined using FlexPepDock. Hit Run button upon providing all the required inputs.



| AutoDock CrankPep: Inputs | |
|---|-------------------------------------|
| Enter the peptide sequence file: | Open |
| Enter the receptor structures: | Open |
| Enter the working directory: | Open |
| 8 number of CPU processors detected in your system | |
| Enter the number of CPU for running jobs in parallel : | 8 |
| 8 Jobs will run in parallel | |
| Enter the number of replicas: | 50 |
| Enter the max steps for one replica: | 2500000 |
| Do you want to run Flexible refinement using FlexPepDock? | <input checked="" type="checkbox"/> |
| Hit Run! | |

9. The PepVis.bash script will be generated in the working directory based on the input provided. The bash script can be run in the terminal by typing `bash PepVis.bash` from the working directory which will perform functions of the provided input and the results will be stored in the working directory.

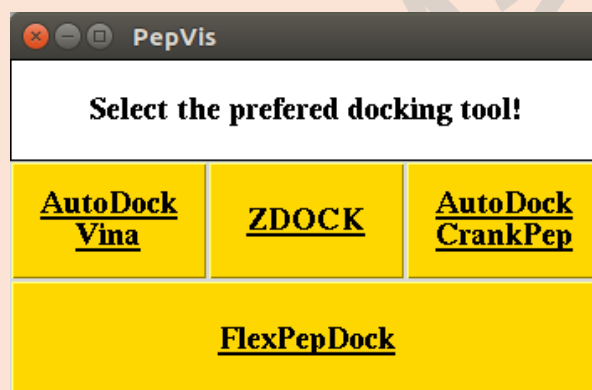


D: Using FlexPepDock for refinement alone

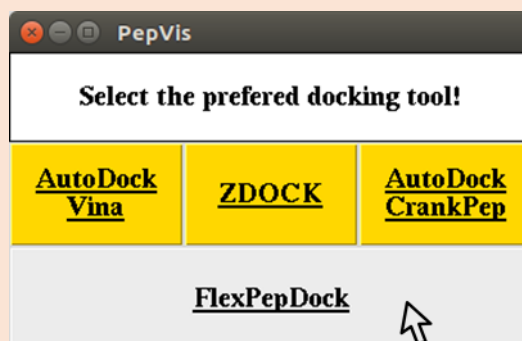
1. Start the script by executing the command `python PepVis.py` in the terminal the main window will pop-up.



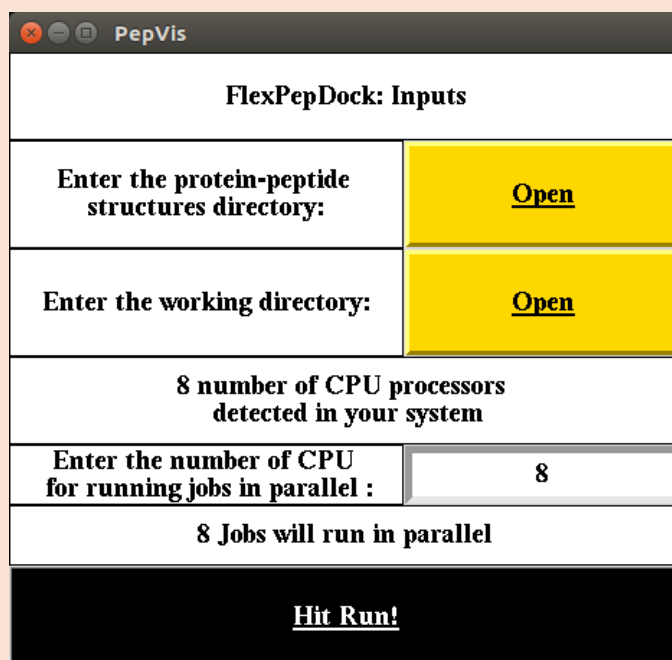
2. The pop-up window with different docking tools or flexible refinement alone buttons will be displayed.



3. Select the FlexPepDock button.



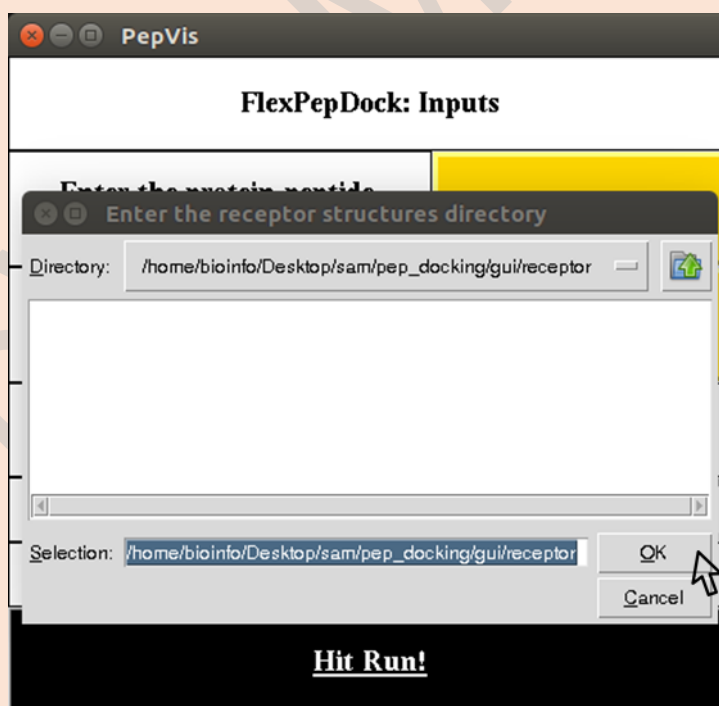
4. A pop-up window will be prompted for obtaining the input for flexible refinement.



The screenshot shows the 'PepVis' application window with the title 'FlexPepDock: Inputs'. It contains several input fields and buttons. The first two rows are for directory selection, each with a yellow 'Open' button. The third row shows the number of CPU processors detected (8). The fourth row is for the number of CPU processors to run in parallel, with a text box containing '8'. The fifth row shows a confirmation message '8 Jobs will run in parallel'. The bottom row is a black bar with the text 'Hit Run!' in white.

| FlexPepDock: Inputs | |
|--|-------------|
| Enter the protein-peptide structures directory: | Open |
| Enter the working directory: | Open |
| 8 number of CPU processors detected in your system | |
| Enter the number of CPU for running jobs in parallel : | 8 |
| 8 Jobs will run in parallel | |
| Hit Run! | |

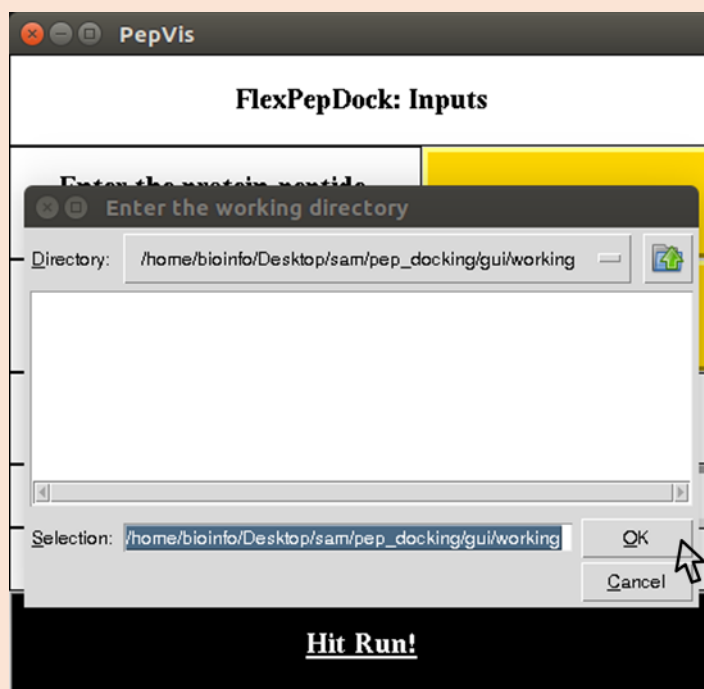
5. Provide the directory path containing protein-peptide complexes to be refined.



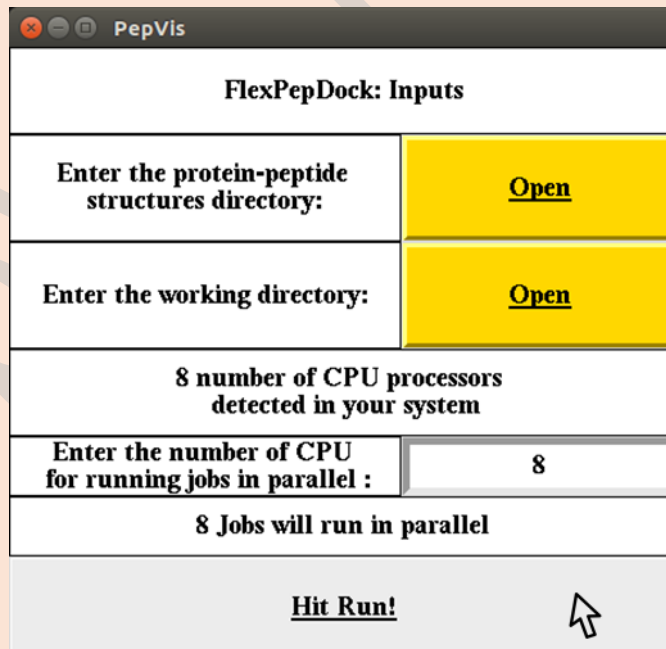
The screenshot shows the 'PepVis' application window with the title 'FlexPepDock: Inputs'. A file selection dialog is open over the first input field. The dialog has a title 'Enter the receptor structures directory' and a text box for the directory path, which is '/home/bioinfo/Desktop/sam/pep_docking/gui/receptor'. There are 'OK' and 'Cancel' buttons at the bottom right of the dialog. The background window shows the same input fields as in the previous screenshot, but the first one is partially obscured by the dialog. The bottom row of the background window is a black bar with the text 'Hit Run!' in white.

| FlexPepDock: Inputs | |
|--|-------------|
| Enter the protein-peptide structures directory: | Open |
| Enter the working directory: | Open |
| 8 number of CPU processors detected in your system | |
| Enter the number of CPU for running jobs in parallel : | 8 |
| 8 Jobs will run in parallel | |
| Hit Run! | |

6. Provide the working directory.



7. Provide the number of CPU's to be used for running in parallel and hit run button.



8. The PepVis.bash script will be generated in the working directory based on the input provided. The bash script can be run in the terminal by typing `bash PepVis.bash` from the working directory which will perform functions of the provided input and the results will be stored in the working directory.

