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 : vumashankar@gmail.com
Email: samdani1593@gmail.com

PepVis Manual

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:

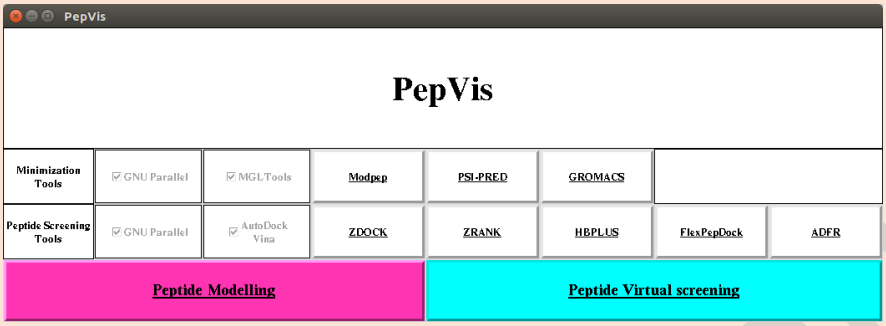
Autodock Vina (http://vina.scripps.edu/download.html**)**
- ✓ vina (/usr/local/bin)
- ✓ vina_split (/usr/local/bin)

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/**)**

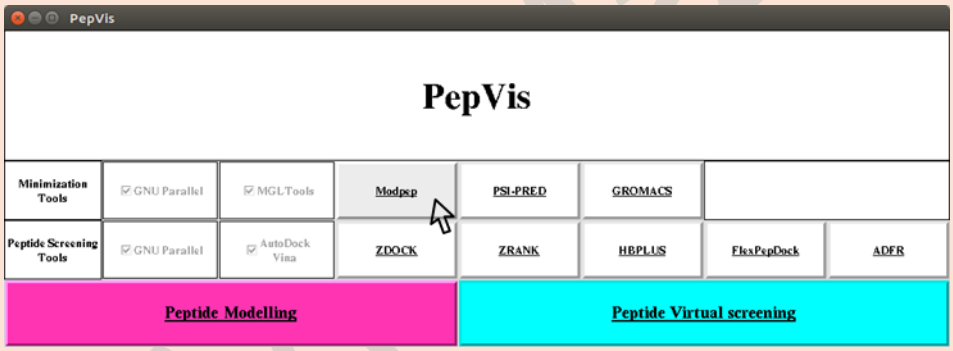
PepVis Manual

Software configuration:

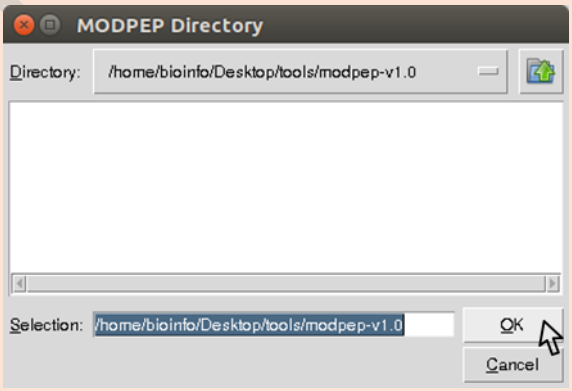
1. On execution of Pepvis, The list of software's configured properly will appear along with a as tick in the checkbox. If you want to configure the tools. For example: ModPep



2. Click the Modpep button

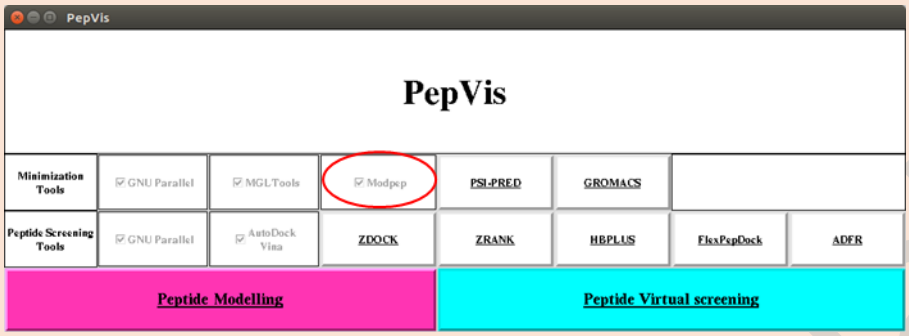


3. Change to the respective directory of where modpep executable was extracted and click ok.

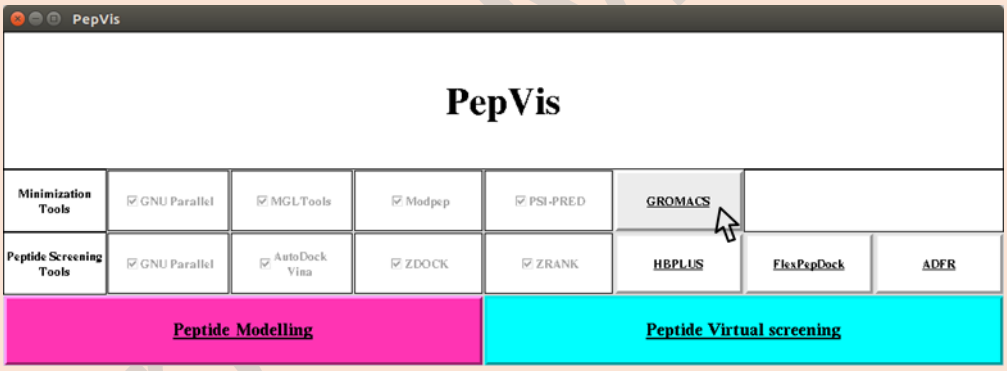


PepVis Manual

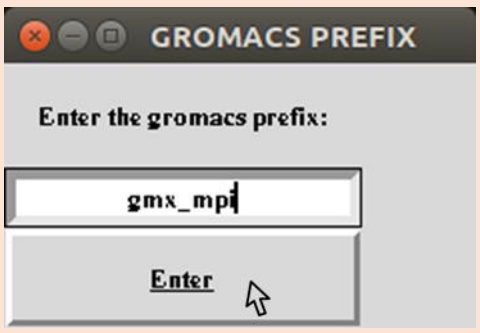
4. If the correct path is provided, the button will be changed to checkbox with tick.



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

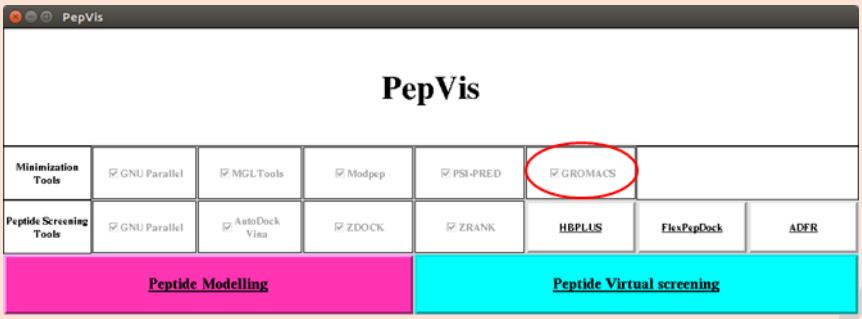


6. A pop-up window will be prompted for obtaining the gromacs prefix like gmx_mpi. Enter the appropriate prefix and click enter button.

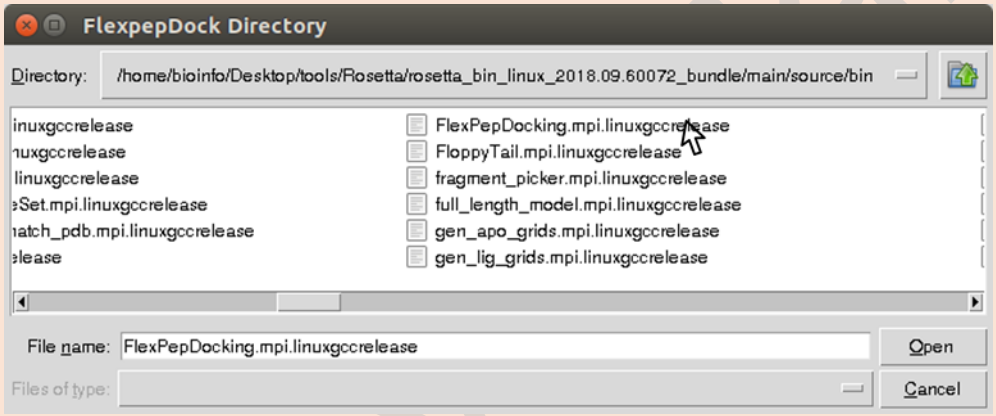


PepVis Manual

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

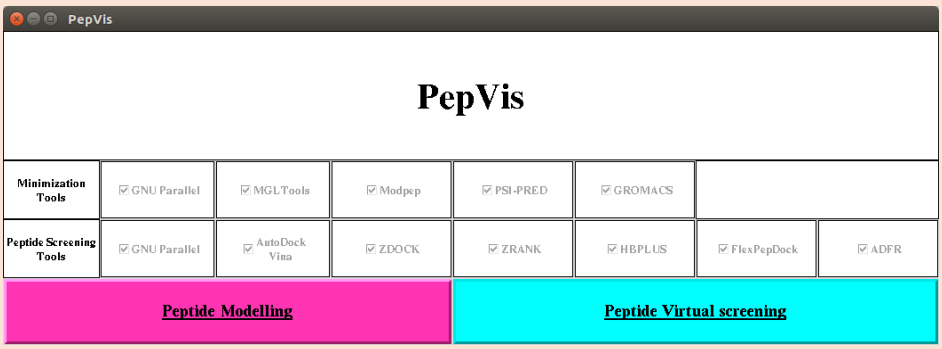


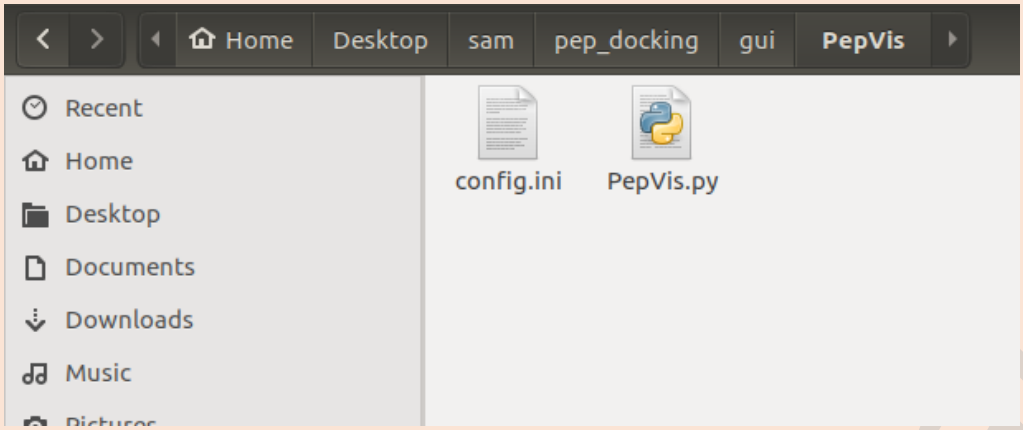
8. In the case of FlexPepDock, go in to the main/source/bin and select the respective FlexPepDocking.mpi.linuxgccrelease exe for configuration.



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

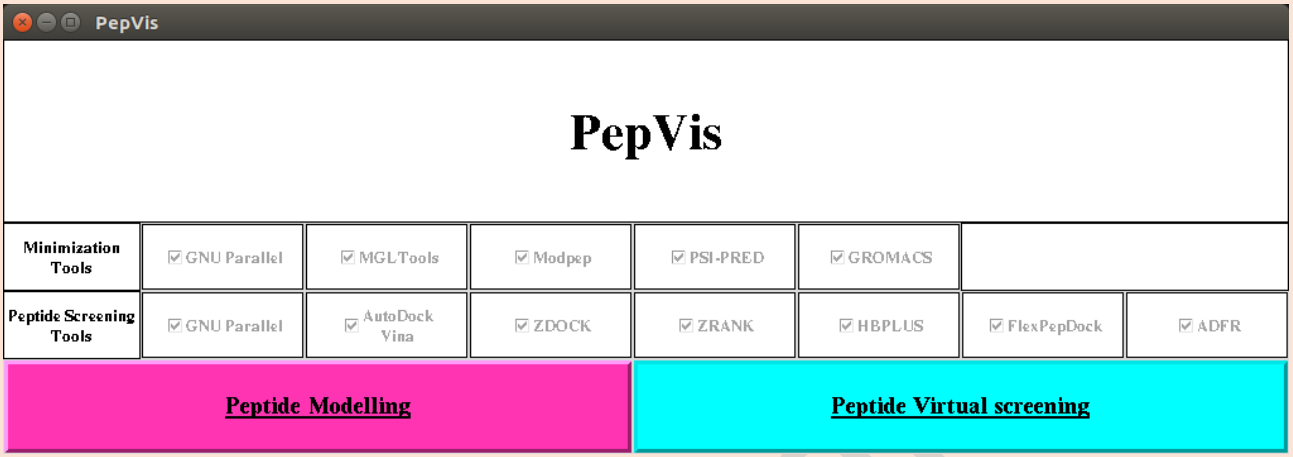
```
[softwares]
MODPEP = /home/bioinfo/Desktop/tools/modpep-v1.0
ZDOCK = /home/bioinfo/Desktop/sam/other/pep_combo/zdock3.0.2_linux_x64
PSIPRED = /home/bioinfo/Desktop/tools/psipred
ZRANK = /home/bioinfo/Desktop/sam/other/pep_combo/zrank2_linux
GROMACS_PREFIX = gmx_mpi
HBPLUS = /home/bioinfo/Desktop/tools/hbplus
FLEXPEPDOCK = /home/bioinfo/Desktop/tools/Rosetta/rosetta_bin_linux_2018.09.60072_bundle/main/source/bin/FlexPepDocking.mpi.linuxgccrelease
ADFRROOT = /home/bioinfo/ADFRsuite-1.0

```

PepVis Manual

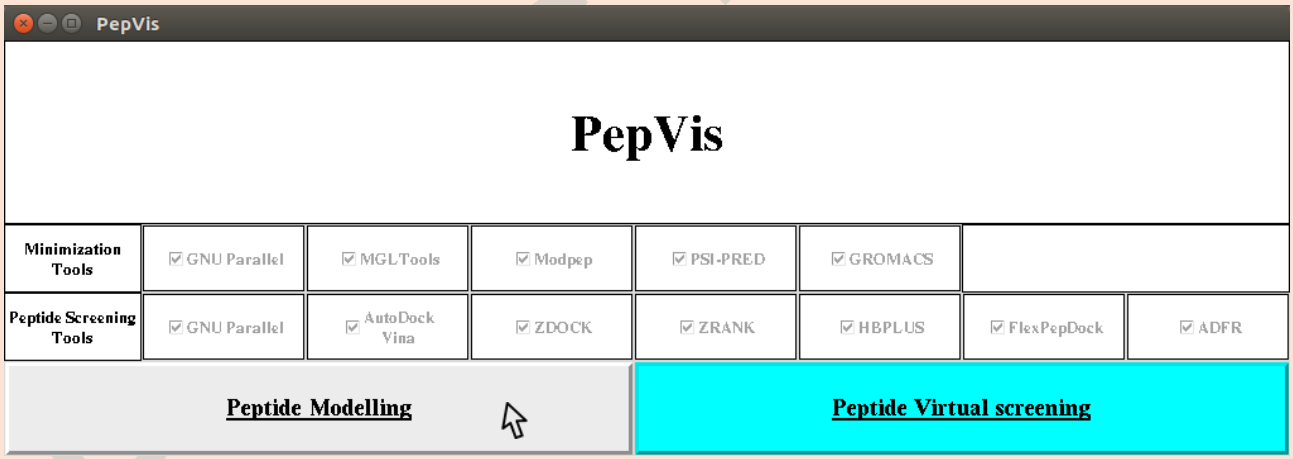
OPERATING INSTRUCTIONS:

Peptide Preparation:

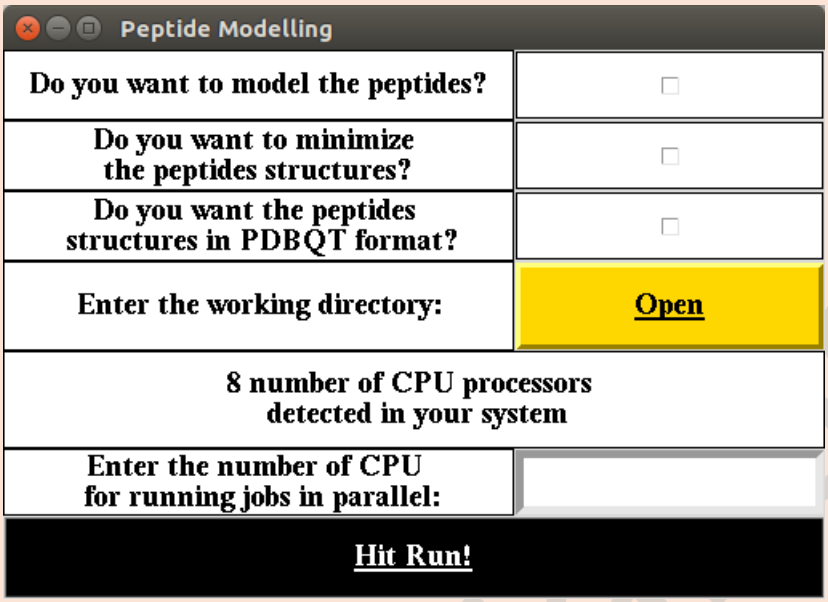
1. After checking the proper installation of softwares, Run the python script PepVis.py in the terminal



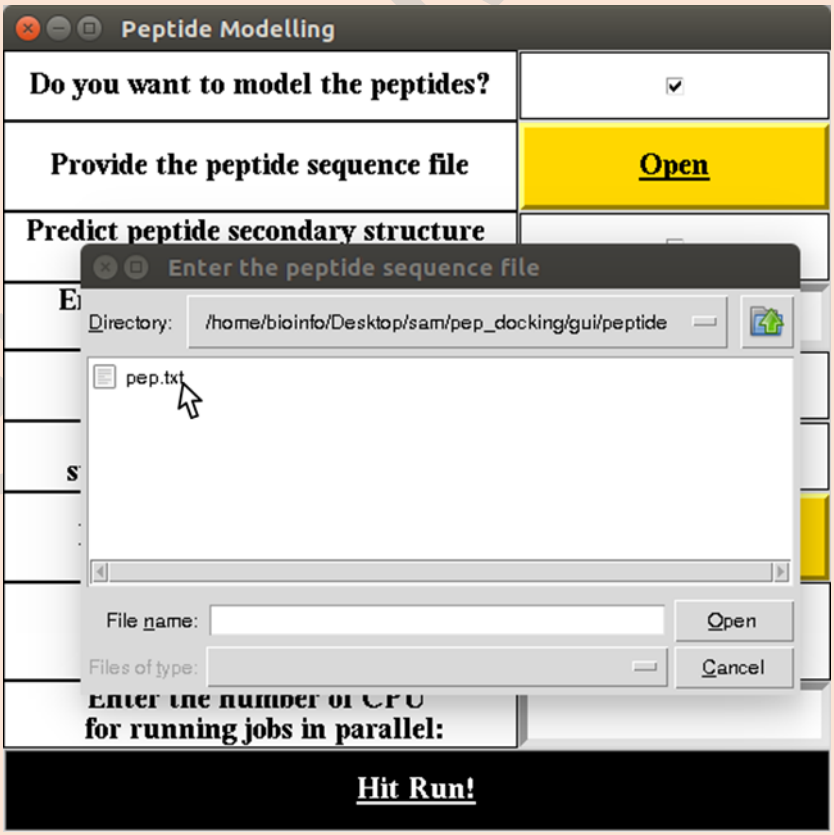
2. Select the Peptide Modelling button



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

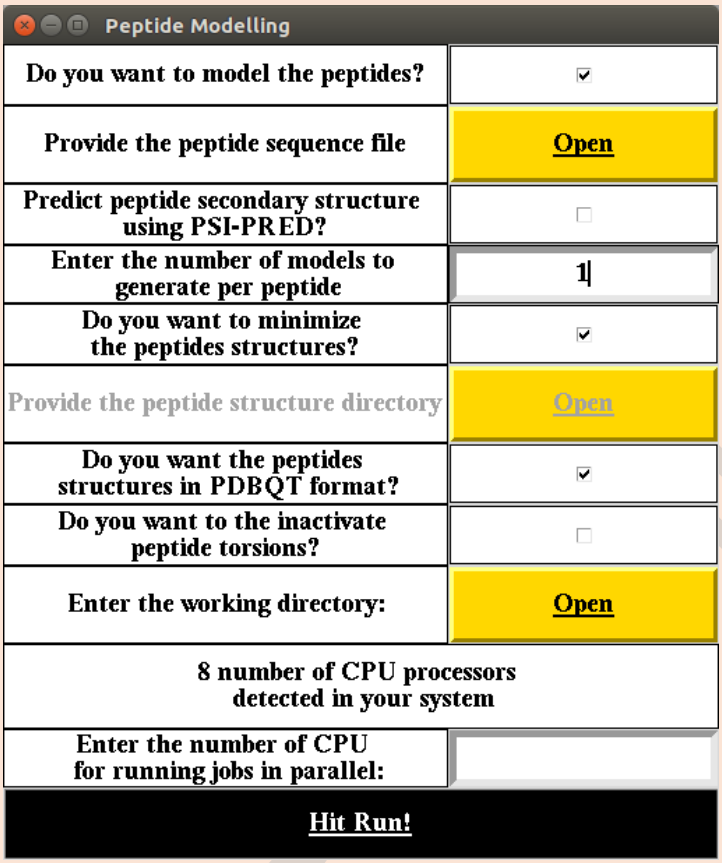


4. Provide the required option in the inputs area:

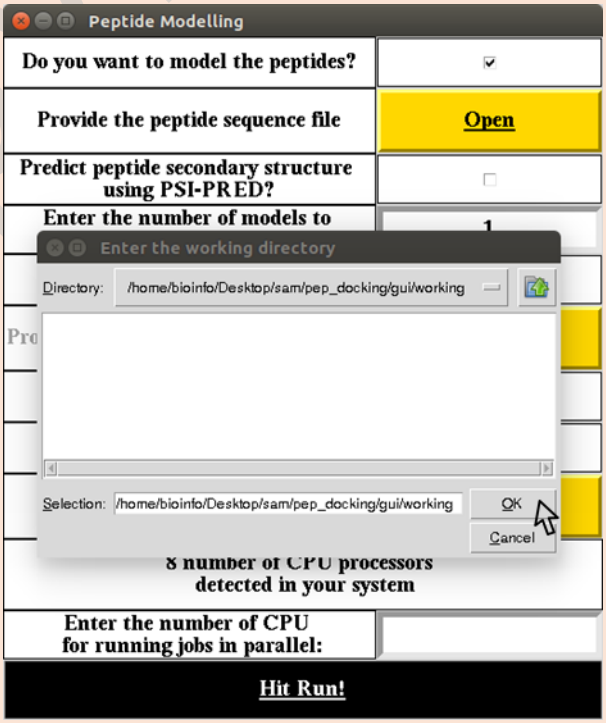


Peptide Manual

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

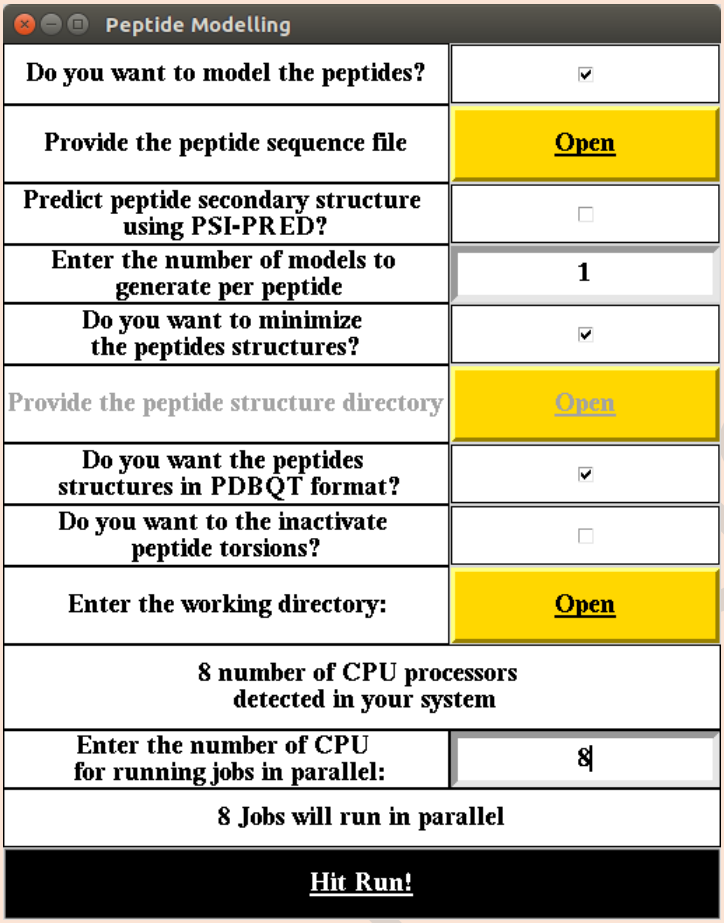


6. Provide the working directory path details

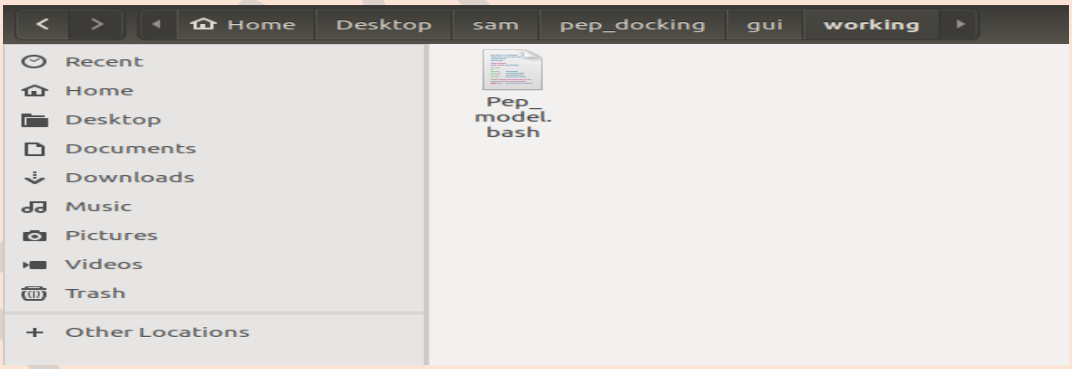


PepVis Manual

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



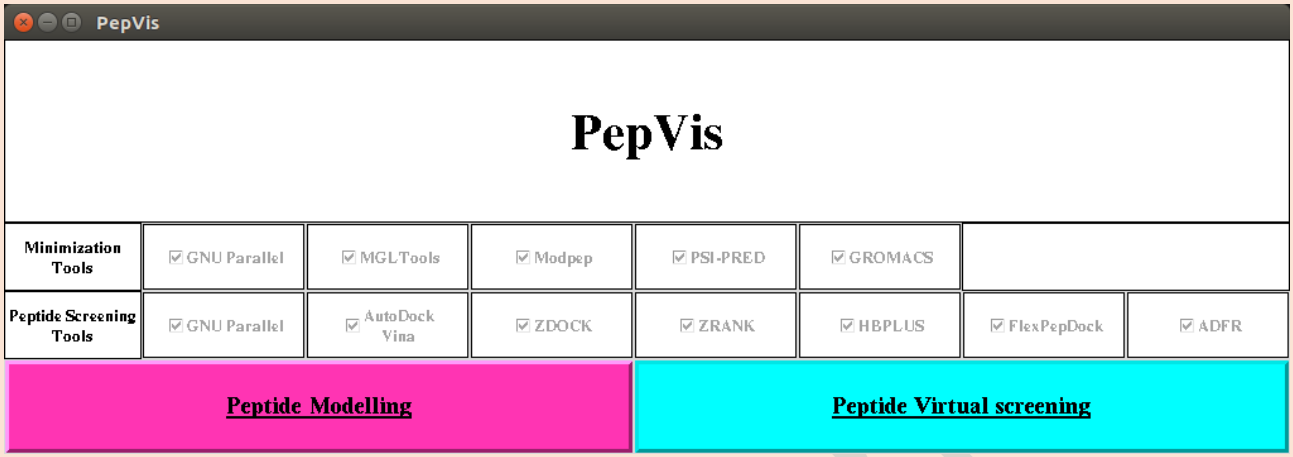
8. Now a bash script will be generated in the working directory provided. The bash script can be run in terminal as bash PepMod.bash alone. The script will run all the provided input steps and peptide datasets will be stored in the working directory.



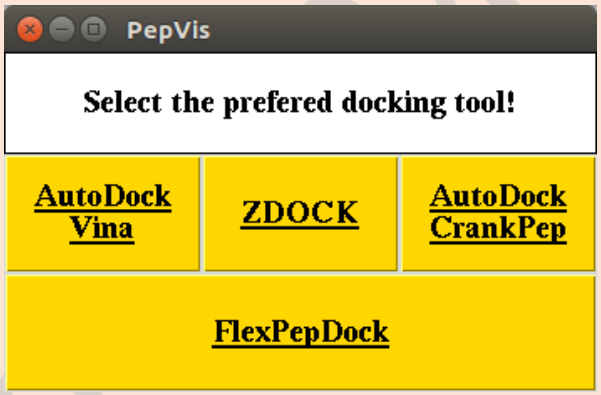
Peptide Virtual Screening:

A: Using AutoDock Vina:

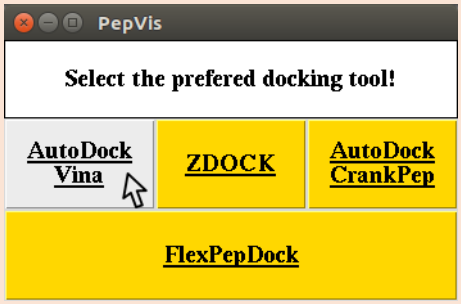
1. Start the Virtual screening script by executing the command python PepVis.py in the terminal the main window will pop-up.



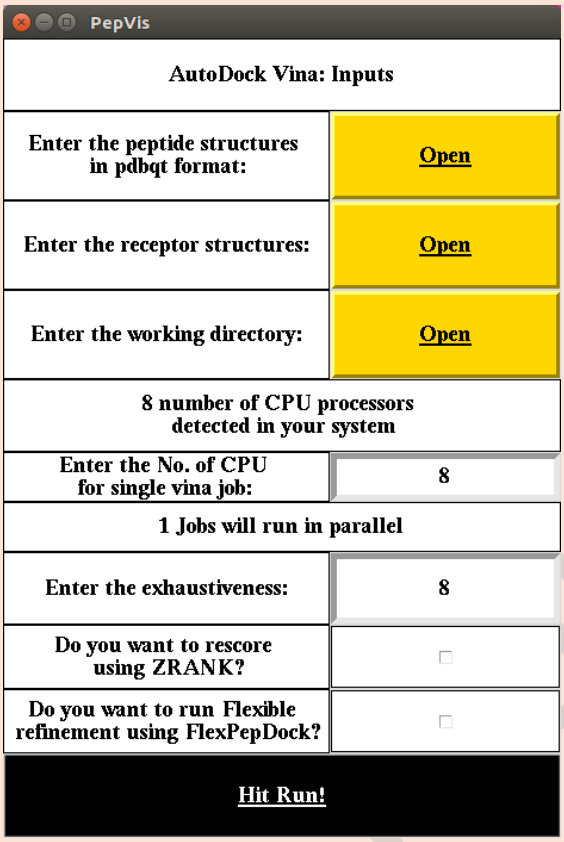
2. The pop-up window with different docking tools or flexible refinement alone buttons will be displayed.



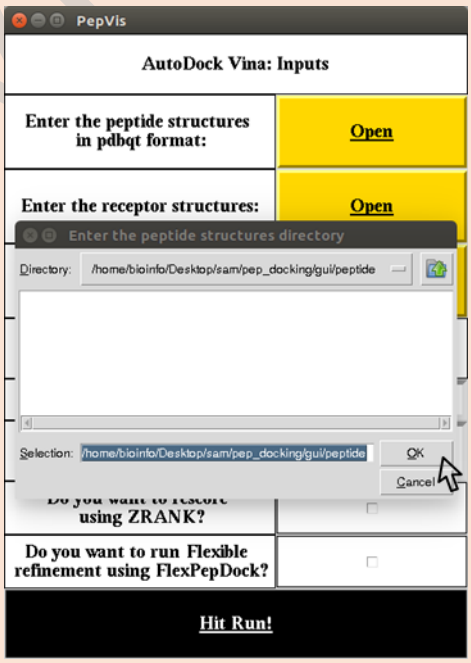
3. Select the required button (AutoDock Vina)



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

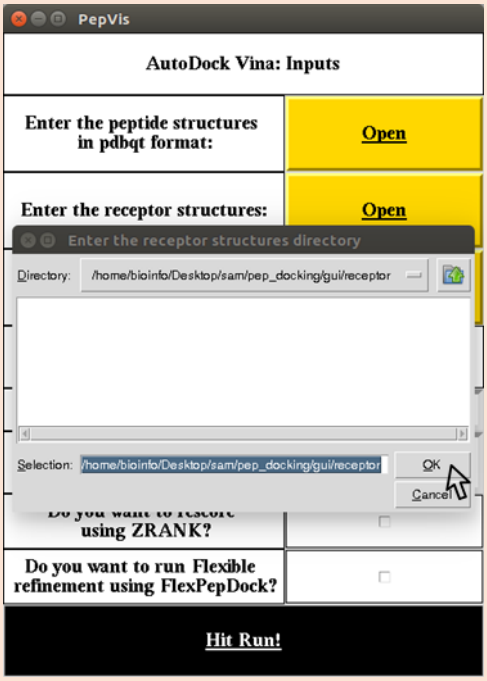


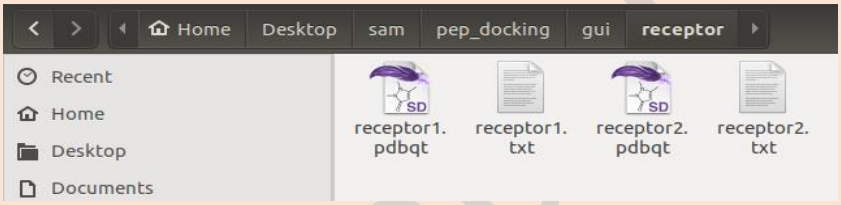
5. Provide the directory containing prepared peptide structures in pdbqt format.



PepVis Manual

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.

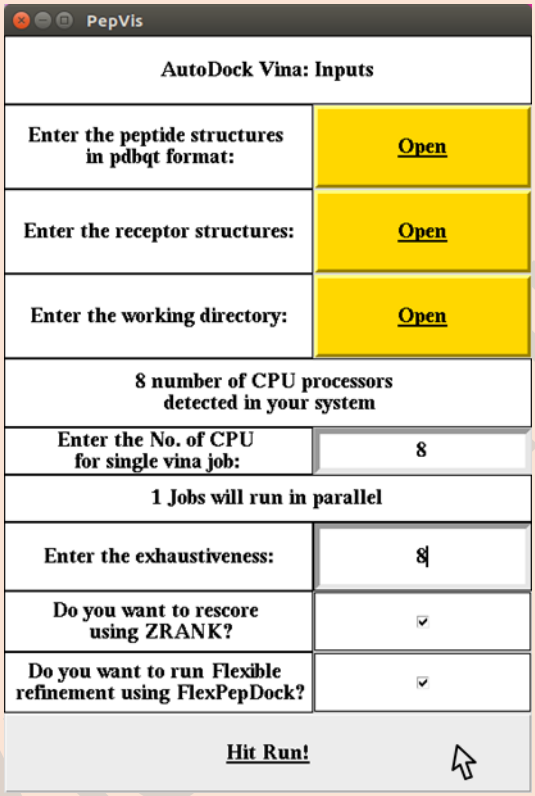




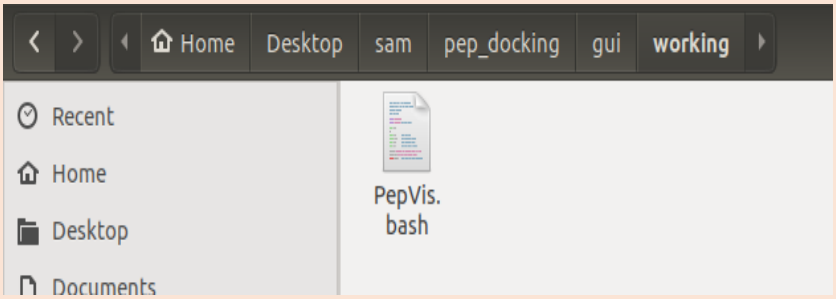
7. Provide the path to working directory.

PepVis Manual

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.



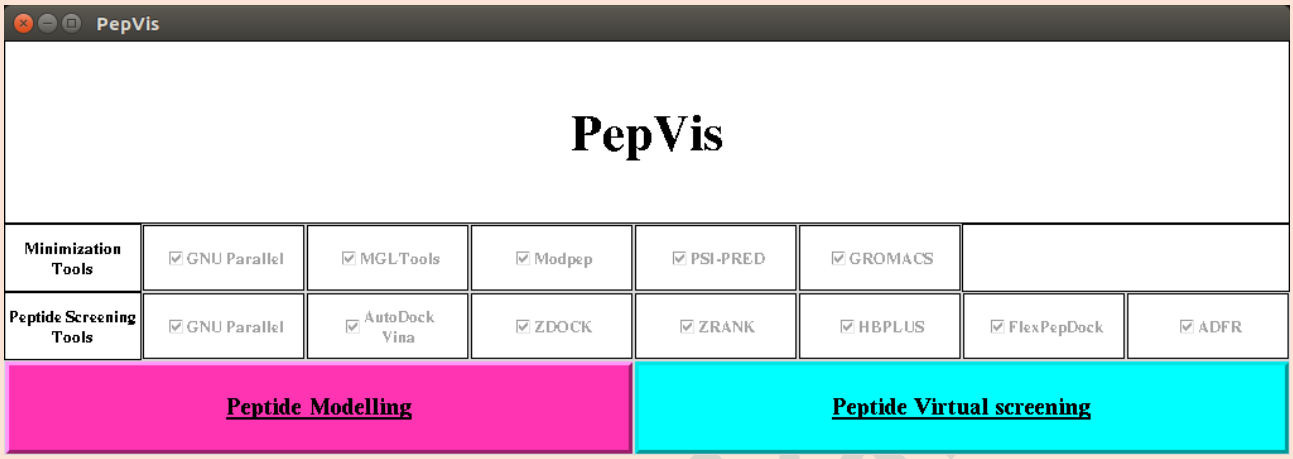
9. The PepVis.bash script will be generated in the working directory based on the input provided. The bash script can be run in the terminal by typing bash PepVis.bash from the working directory which will perform functions of the provided input and the results will be stored in the working directory.



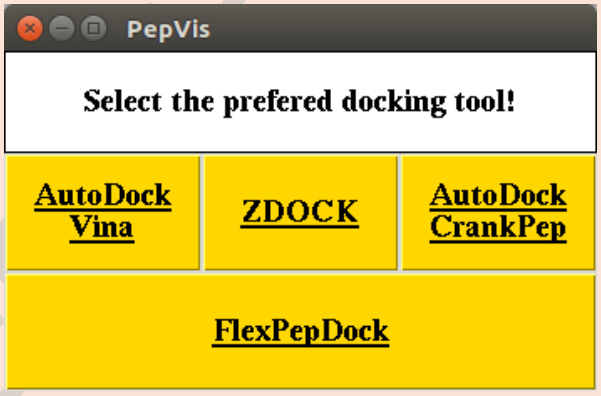
PepVis Manual

B: Using ZDOCK:

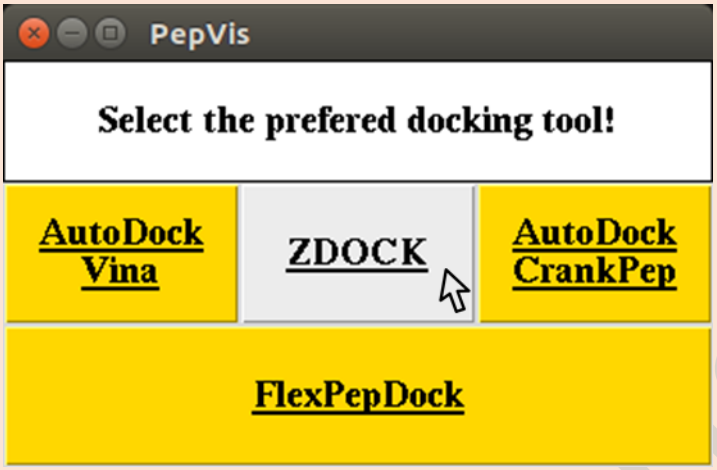
1. Start the script by executing the command python PepVis.py in the terminal and the main window will pop-up.



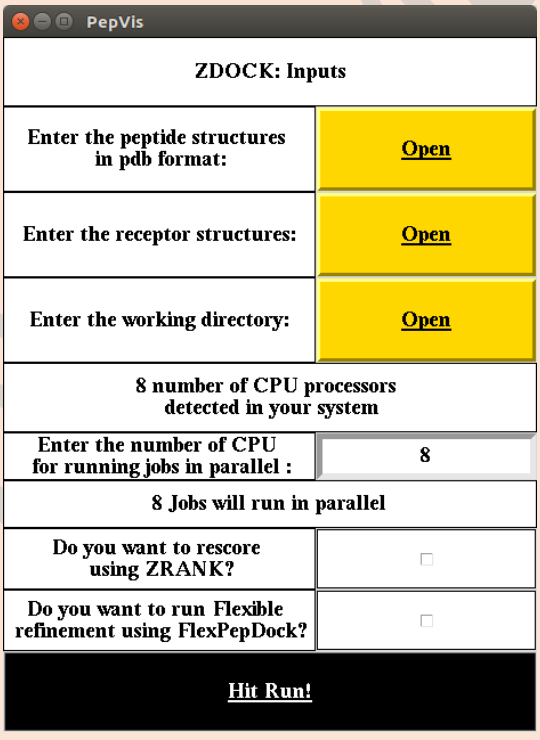
2. The pop-up window with different docking tools or flexible refinement alone buttons will be displayed.



3. Select the ZDOCK button.

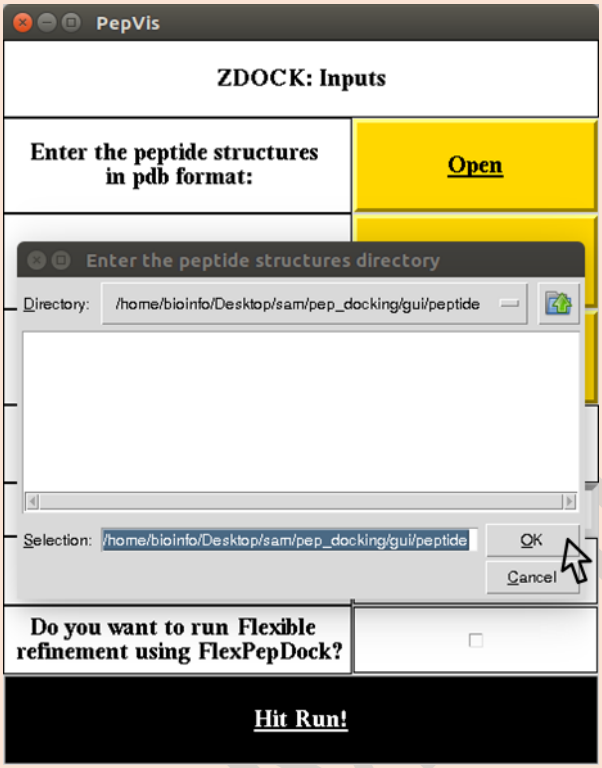


4. The pop-up window for obtaining the ZDOCK input will be prompted.



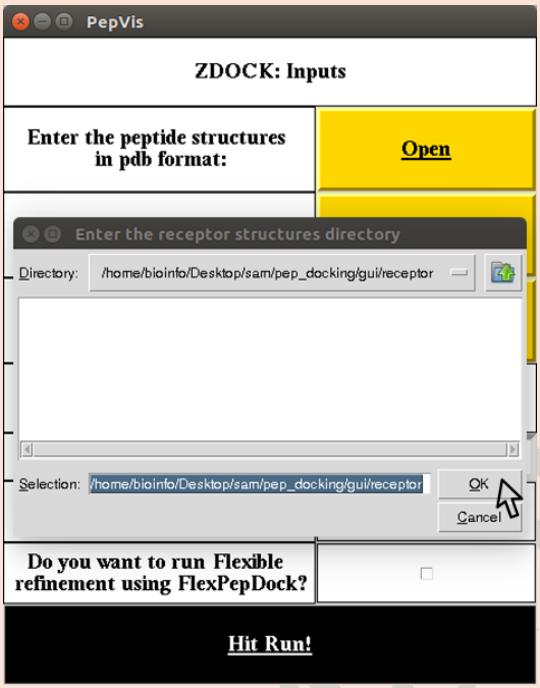
PepVis Manual

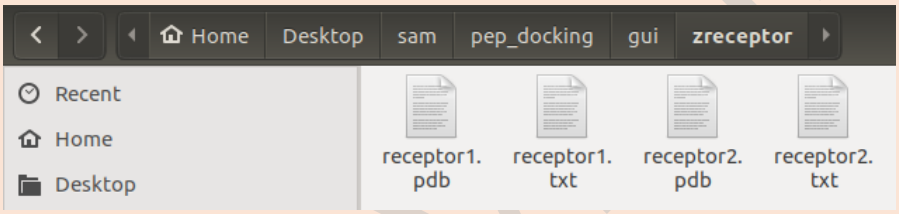
5. Provide the directory containing prepared peptide structures in .pdb format

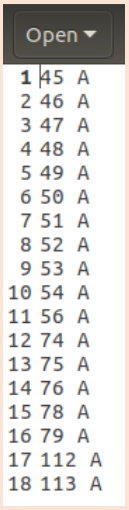


PepVis Manual

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.

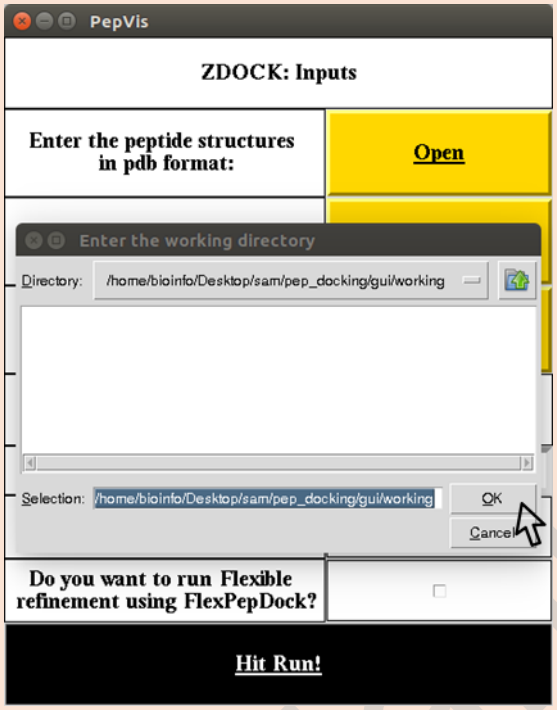




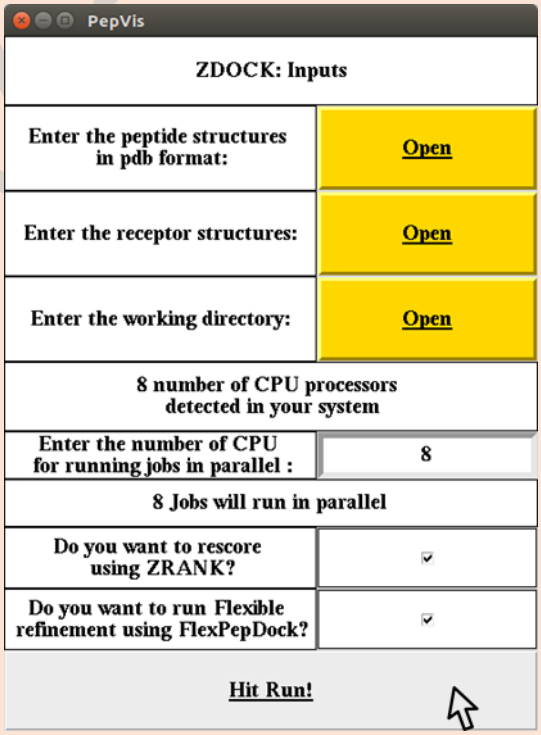


PepVis Manual

7. Provide the path to working directory.

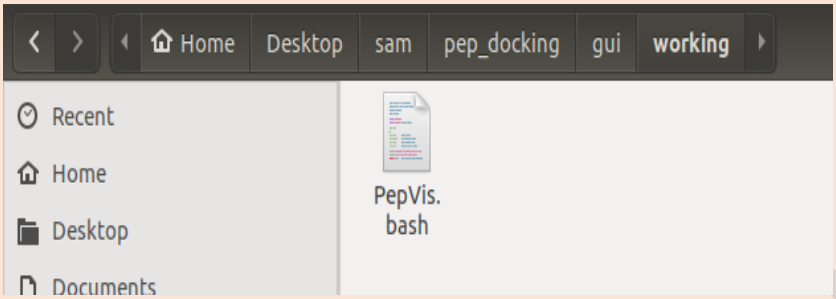


8. Provide the number of CPU's to run in parallel. Check the button if Rescore using ZRANK or not? If the Flexible refinement of the complex need to be performed?



PepVis Manual

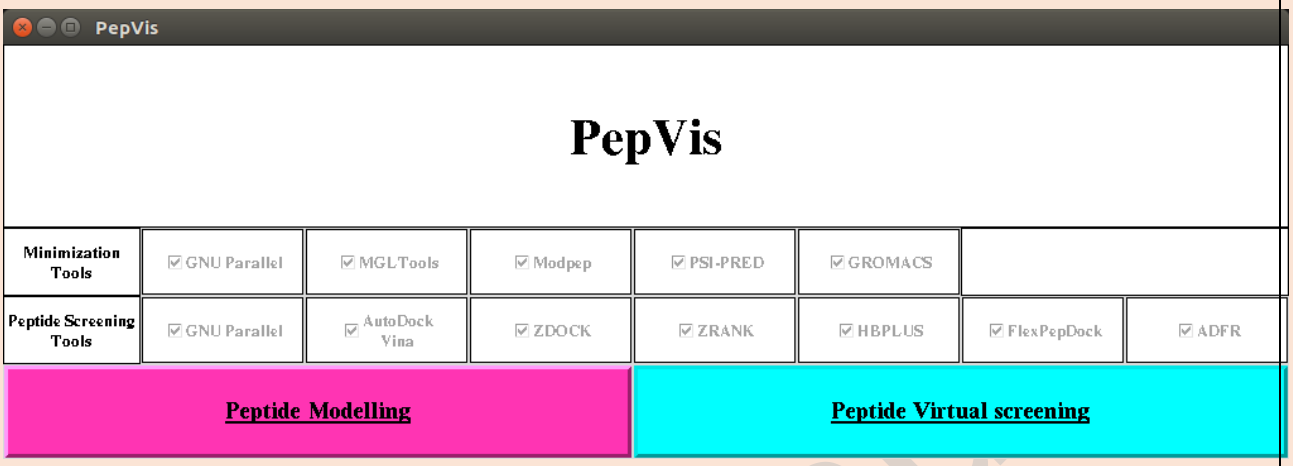
9. The PepVis.bash script will be generated in the working directory based on the input provided. The bash script can be run in the terminal by typing bash PepVis.bash from the working directory which will perform functions of the provided input and the results will be stored in the working directory.



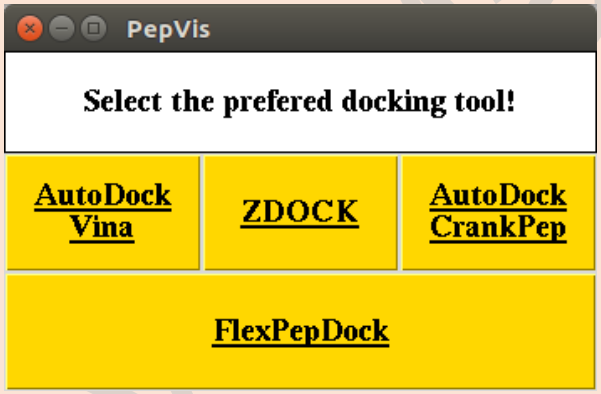
PepVis Manual

C: Using ADCP:

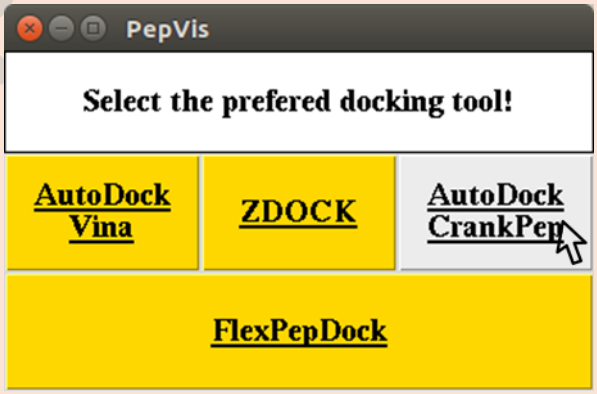
1. Start the script by executing the command python PepVis.py in the terminal and the main window will pop-up.



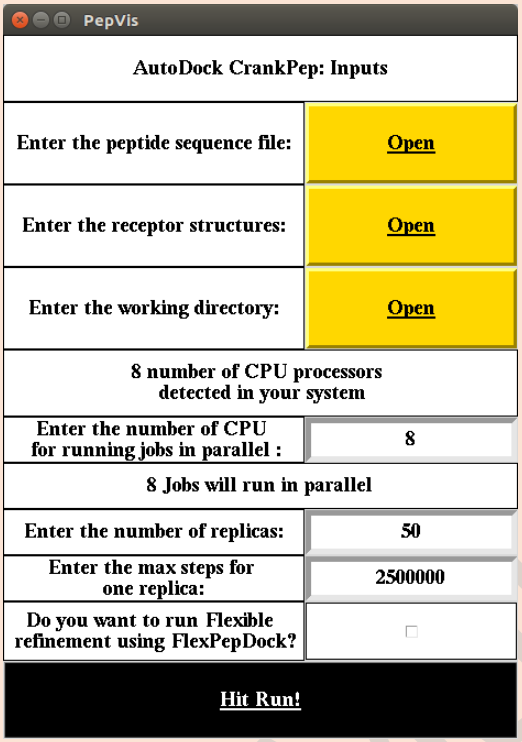
2. The pop-up window with different docking tools or flexible refinement alone buttons will be displayed.



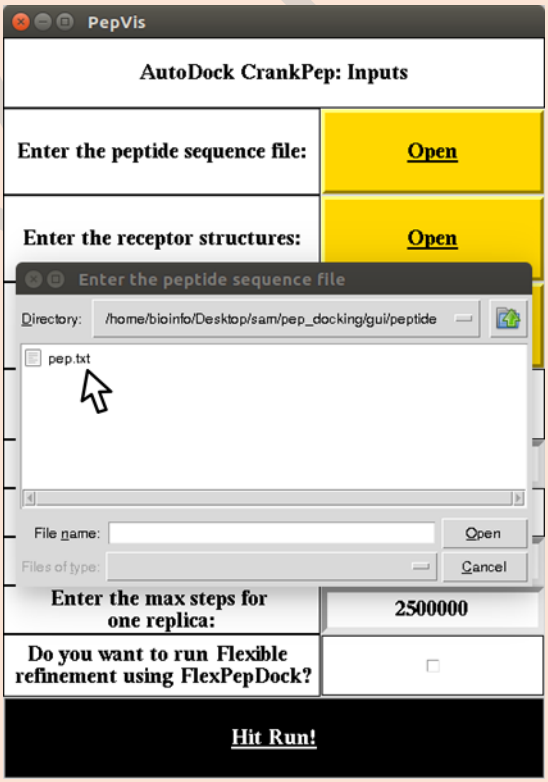
3. Select the AutoDock CrankPep button.



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

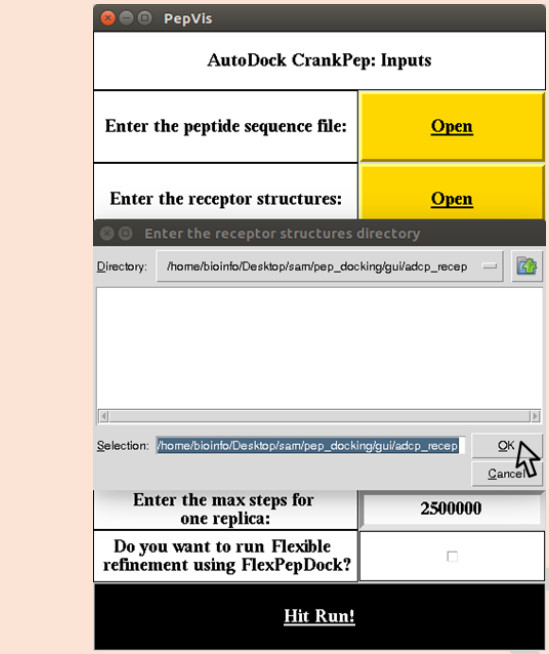


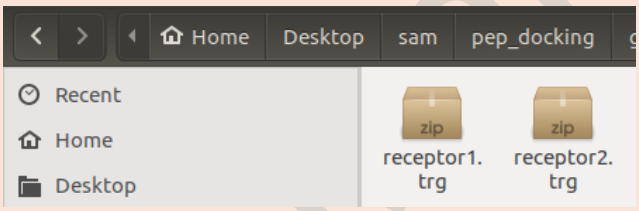
5. Provide the peptide sequences containing text file.



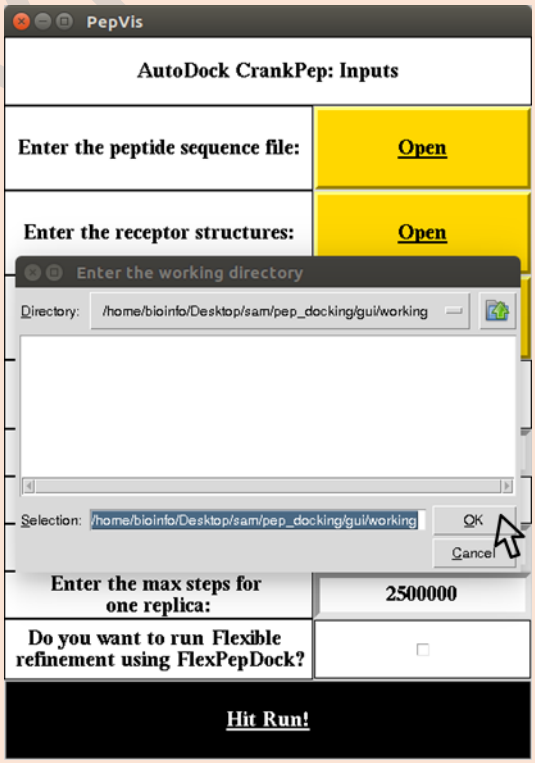
PepVis Manual

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



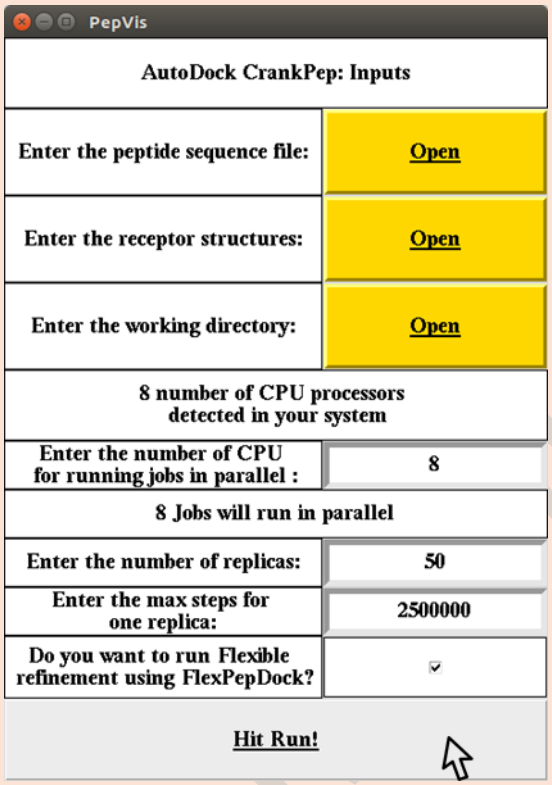


7. Provide the working directory.

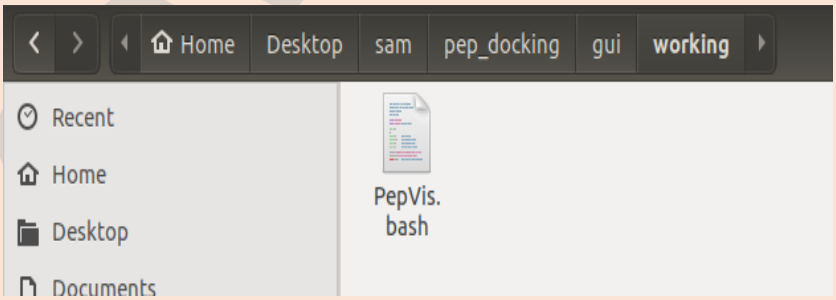


PepVis Manual

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.



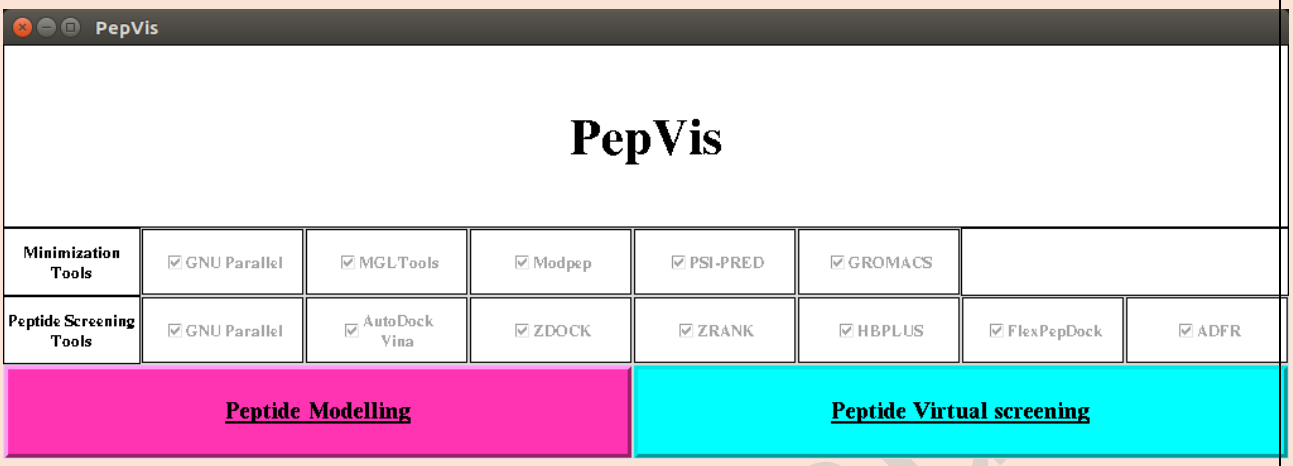
9. The PepVis.bash script will be generated in the working directory based on the input provided. The bash script can be run in the terminal by typing bash PepVis.bash from the working directory which will perform functions of the provided input and the results will be stored in the working directory.



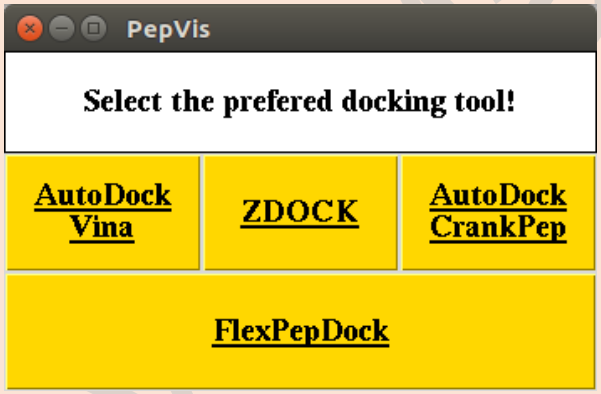
PepVis Manual

D: Using FlexPepDock for refinement alone

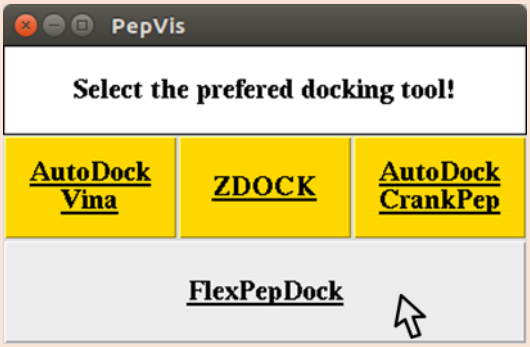
1. Start the script by executing the command python PepVis.py in the terminal the main window will pop-up.



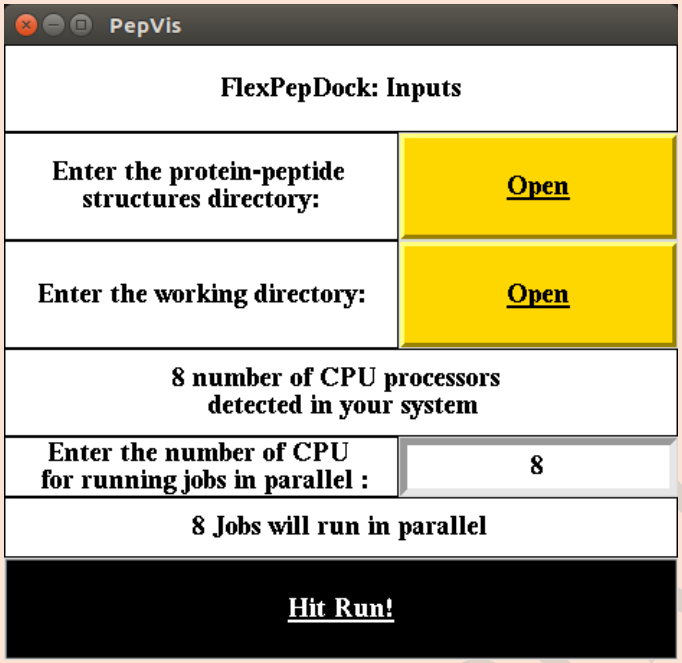
2. The pop-up window with different docking tools or flexible refinement alone buttons will be displayed.



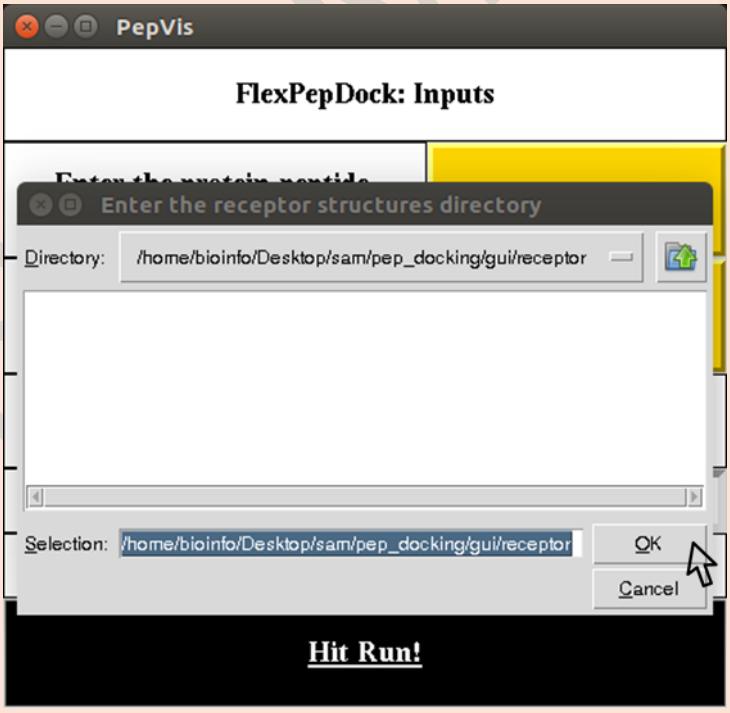
3. Select the FlexPepDock button.



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

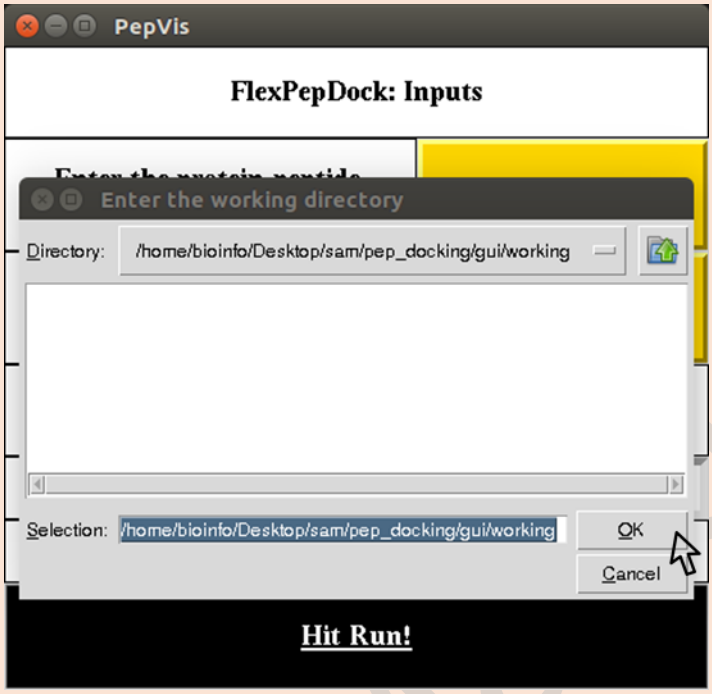


5. Provide the directory path containing protein-peptide complexes to be refined.

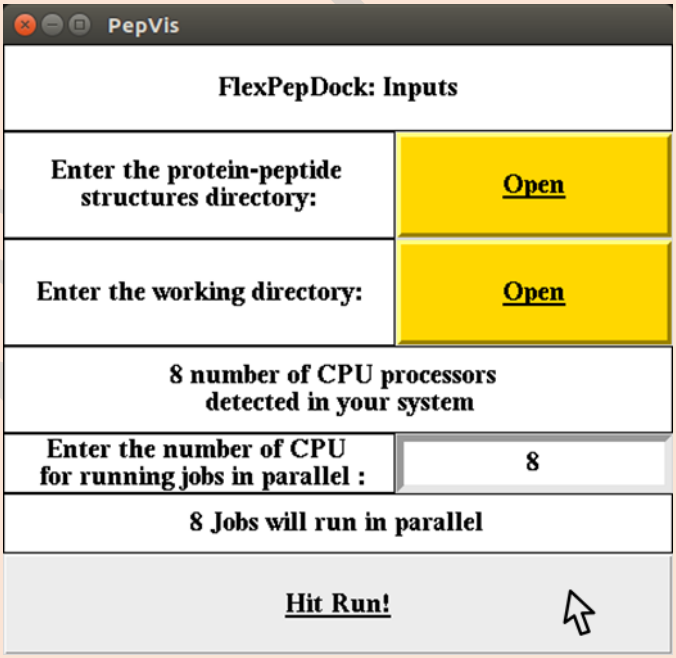


PepVis Manual

6. Provide the working directory.

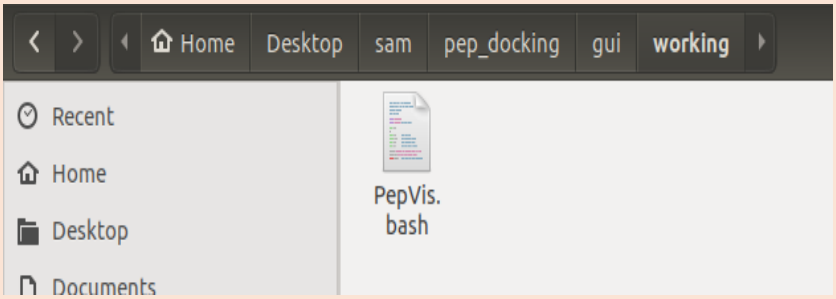


7. Provide the number of CPU's to be used for running in parallel and hit run button.



PepVis Manual

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.



PepVis Manual