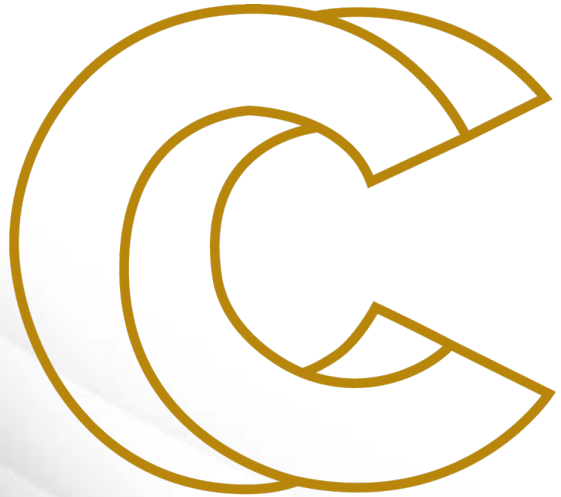


EURO^2

High throughput Virtual Screening with AutoDOCK vina program

Asst.Prof Sefer BADAY

İTÜ Faculty of Computer and Informatics Engineering

ncc@ulakbim.gov.tr

Lesson 4: Analysis of the results

- Outputs folder contains docking results for each task

```
analysis  part1   part11  part3  part5  part7  part9          prepare_jobs.py
outputs   part10  part2   part4  part6  part8  prep_materials  submit_jobs.sh

[sbaday@altay:~/hts_dockings/5ht_1b ]$ cd outputs/

[sbaday@altay:~/hts_dockings/5ht_1b/outputs ]$ ls
task100_output  task13_output   task179_output  task217_output  task256_output  task295_output  task333_output  task64_output
task101_output  task140_output  task17_output   task218_output  task257_output  task296_output  task334_output  task65_output
task102_output  task141_output  task180_output  task219_output  task258_output  task297_output  task335_output  task66_output
task103_output  task142_output  task181_output  task21_output   task259_output  task298_output  task336_output  task67_output
task104_output  task143_output  task182_output  task220_output  task25_output   task299_output  task337_output  task68_output
task105_output  task144_output  task183_output  task221_output  task260_output  task29_output   task338_output  task69_output
task106_output  task145_output  task184_output  task222_output  task261_output  task2_output    task339_output  task6_output
task107_output  task146_output  task185_output  task223_output  task262_output  task300_output  task33_output   task70_output
task108_output  task147_output  task186_output  task224_output  task263_output  task301_output  task340_output  task71_output
task109_output  task148_output  task187_output  task225_output  task264_output  task302_output  task341_output  task72_output
```

Lesson 4: Analysis of the results

- In each task folder we have binding pose as pdbqt format and
- Log file showing the docking score

```
[sbaday@altay:~/hts_dockings/5ht_1b/outputs ]$ cd task100_output

[sbaday@altay:~/hts_dockings/5ht_1b/outputs/task100_output ]$ ls
ZINC000000002075_log.txt    ZINC000013530263_out.pdbqt  ZINC000072225704_out.pdbqt  ZINC000218698569_log.txt
ZINC000000002075_out.pdbqt  ZINC000013532044_log.txt    ZINC000072225883_log.txt    ZINC000218698569_out.pdbqt
ZINC000000004785_log.txt    ZINC000013532044_out.pdbqt  ZINC000072225883_out.pdbqt  ZINC000218704204_log.txt
ZINC000000004785_out.pdbqt  ZINC000013547036_log.txt    ZINC000072226338_log.txt    ZINC000218704204_out.pdbqt
ZINC000000009064_log.txt    ZINC000013547036_out.pdbqt  ZINC000072226338_out.pdbqt  ZINC000218752408_log.txt
ZINC000000009064_out.pdbqt  ZINC000013547573_log.txt    ZINC000072227114_log.txt    ZINC000218752408_out.pdbqt
ZINC000000027951_log.txt    ZINC000013547573_out.pdbqt  ZINC000072227114_out.pdbqt  ZINC000218839496_log.txt
ZINC000000027951_out.pdbqt  ZINC000013552585_log.txt    ZINC000072227140_log.txt    ZINC000218839496_out.pdbqt
ZINC000000029106_log.txt    ZINC000013552585_out.pdbqt  ZINC000072227140_out.pdbqt  ZINC000218840208_log.txt
ZINC000000029106_out.pdbqt  ZINC000013552969_log.txt    ZINC000072227210_log.txt    ZINC000218840208_out.pdbqt
ZINC000000029328_log.txt    ZINC000013552969_out.pdbqt  ZINC000072227210_out.pdbqt  ZINC000218841070_log.txt
```

Lesson 4: extracting docking scores

```
[sbaday@altay:~/hts_dockings/5ht_1b/outputs/task100_output ]$ cat ZINC000000002075_log.txt
#################################################################
# If you used AutoDock Vina in your work, please cite:          #
#                                                               #
# O. Trott, A. J. Olson,                                        #
# AutoDock Vina: improving the speed and accuracy of docking    #
# with a new scoring function, efficient optimization and       #
# multithreading, Journal of Computational Chemistry 31 (2010)  #
# 455-461                                                       #
#                                                               #
# DOI 10.1002/jcc.21334                                         #
#                                                               #
# Please see http://vina.scripps.edu for more information.      #
#################################################################

Reading input ... done.
Setting up the scoring function ... done.
Analyzing the binding site ... done.
Using random seed: 1365352780
Performing search ... done.
Refining results ... done.

mode |   affinity | dist from best mode
     | (kcal/mol) | rmsd l.b.| rmsd u.b.
-----+------------+----------+----------
   1        -7.7      0.000      0.000
   2        -7.3      3.305      4.889
   3        -7.3      2.412      4.093
   4        -7.2      1.846      2.772
   5        -7.0      1.450      2.296
   6        -6.9      2.796      4.447
   7        -6.9      1.625      5.655
   8        -6.7      5.755      6.648
   9        -6.7      2.691      4.380
Writing output ... done.
```

- A sample output
- We need to get docking score as -7.7 from this file

Lesson 4: extracting docking scores

- We create a separate folder for analysis.
- Then we will execute calc_all_docking_scores.py which goes through all log files and exract docking scores.

```
[sbaday@altay:~/hts_dockings/5ht_1b/analysis ]$ cat calc_all_docking_scores.py
docking_results="/ari/users/sbaday/hts_dockings/5ht_1b/outputs"
output_filename="/ari/users/sbaday/hts_dockings/5ht_1b/analysis/all_docking_results.csv"
import os
os.makedirs("analysis",exist_ok=True)
output_file=open(output_filename,"w")

def get_top_score(filename):
    lines=open(filename).readlines()
    for line in lines:
        if line.startswith("   1 "):
            score=line.split()[1]
            return score

import os
folders=os.listdir(docking_results)

count=0
for folder in folders:
    for file in os.listdir(os.path.join(docking_results,folder)):
        if file.endswith(".txt"):
            score=get_top_score(os.path.join(docking_results,folder,file))
            if score is not None:
                #output_file.write(file.strip("_log.txt")+" " +score+"\n")
                output_file.write(os.path.join(docking_results,folder) +" "+file.split("_")[0] + " " +score +"\n")

output_file.close()
```

Lesson 4: extracting docking scores

- The content of calc_all_docking_scores.py file.

```
docking_results="/ari/users/sbaday/hts_dockings/5ht_1b/outputs"
output_filename="/ari/users/sbaday/hts_dockings/5ht_1b/analysis/all_docking_results.csv"
import os
os.makedirs("analysis",exist_ok=True)
output_file=open(output_filename,"w")


def get_top_score(filename):
    lines=open(filename).readlines()
    for line in lines:
        if line.startswith("   1 "):
            score=line.split()[1]
            return score


import os
folders=os.listdir(docking_results)


count=0
for folder in folders:
    for file in os.listdir(os.path.join(docking_results,folder)):
        if file.endswith(".txt"):
            score=get_top_score(os.path.join(docking_results,folder,file))
            if score is not None:
                #output_file.write(file.strip("_log.txt")+" " +score+"\n")
                output_file.write(os.path.join(docking_results,folder) +" "+file.split("_")[0] + " " +score +"\n")


output_file.close()
```

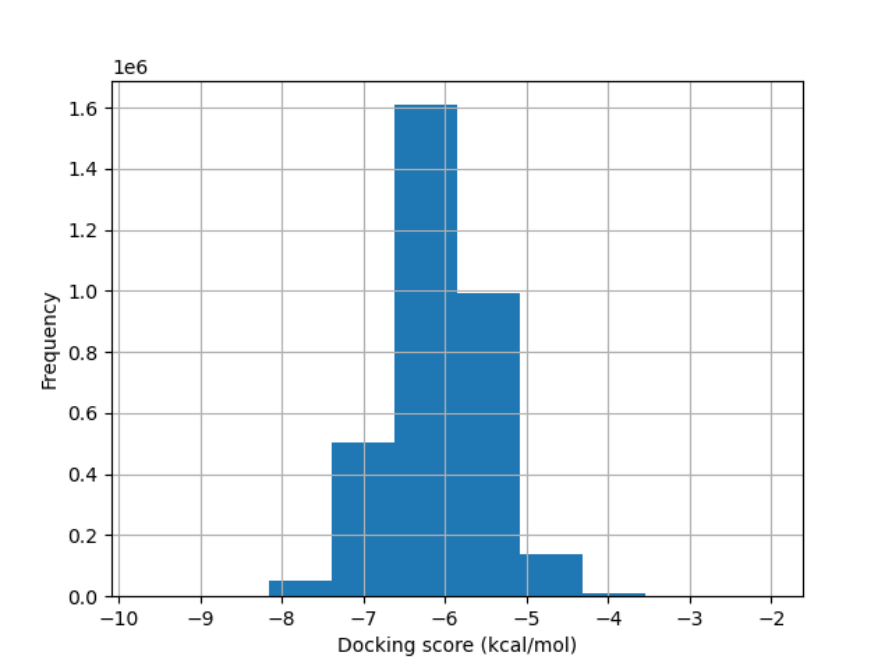
Lesson 4: extracting docking scores

- Preview of **all_docking_results.csv** file

```
[sbaday@altay:~/hts_dockings/5ht_1b/analysis ]$ cat all_docking_results.csv
/ari/users/sbaday/hts_dockings/5ht_1b/outputs/task100_output,ZINC000000002075,-7.7
/ari/users/sbaday/hts_dockings/5ht_1b/outputs/task100_output,ZINC000000009064,-9.6
/ari/users/sbaday/hts_dockings/5ht_1b/outputs/task100_output,ZINC000000029328,-7.4
/ari/users/sbaday/hts_dockings/5ht_1b/outputs/task100_output,ZINC000000029370,-6.9
/ari/users/sbaday/hts_dockings/5ht_1b/outputs/task100_output,ZINC000000029396,-8.9
/ari/users/sbaday/hts_dockings/5ht_1b/outputs/task100_output,ZINC000000029770,-8.7
/ari/users/sbaday/hts_dockings/5ht_1b/outputs/task100_output,ZINC000000027951,-6.8
/ari/users/sbaday/hts_dockings/5ht_1b/outputs/task100_output,ZINC000000029106,-9.0
/ari/users/sbaday/hts_dockings/5ht_1b/outputs/task100_output,ZINC000000034211,-8.1
/ari/users/sbaday/hts_dockings/5ht_1b/outputs/task100_output,ZINC000000035634,-7.5
/ari/users/sbaday/hts_dockings/5ht_1b/outputs/task100_output,ZINC000000029663,-7.8
/ari/users/sbaday/hts_dockings/5ht_1b/outputs/task100_output,ZINC000000034247,-9.1
/ari/users/sbaday/hts_dockings/5ht_1b/outputs/task100_output,ZINC000000036679,-6.2
/ari/users/sbaday/hts_dockings/5ht_1b/outputs/task100_output,ZINC000000036973,-8.6
/ari/users/sbaday/hts_dockings/5ht_1b/outputs/task100_output,ZINC000000037608,-9.2
/ari/users/sbaday/hts_dockings/5ht_1b/outputs/task100_output,ZINC000000038721,-7.8
```

Lesson 4: getting binding pose of top ranked molecules

- This python scripts will get the pdbqt result for molecules ranked top-100.
- The script also produces docking score distribution.




```
import pandas as pd
import matplotlib.pyplot as plt
import shutil
import os
data=pd.read_csv("all_docking_results.csv",delim_whitespace=True,header=None,names=["path","sequence","score"])
data.sort_values(by="score",inplace=True)
# create histogram of docking scores
import matplotlib.pyplot as plt
data.score.hist()
plt.xlabel("Docking score (kcal/mol)")
plt.ylabel("Frequency")
plt.savefig("docking_score_distribution.png")

# get top n-ranked sequence
n=100
for i in range(n):
    path=data.path.iloc[i]
    seq=data.sequence.iloc[i]
    score=data.score.iloc[i]
    pdbqt_path=os.path.join(path,seq)+"_out.pdbqt"
    output_path="top_"+str(i+1)+"_"+seq+"_"+str(score)+".pdbqt"
    shutil.copyfile(pdbqt_path,output_path)
```

Thanks.

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101101903. The JU receives support from the Digital Europe Programme and Germany, Bulgaria, Austria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Greece, Hungary, Ireland, Italy, Lithuania, Latvia, Poland, Portugal, Romania, Slovenia, Spain, Sweden, France, Netherlands, Belgium, Luxembourg, Slovakia, Norway, Türkiye, Republic of North Macedonia, Iceland, Montenegro, Serbia