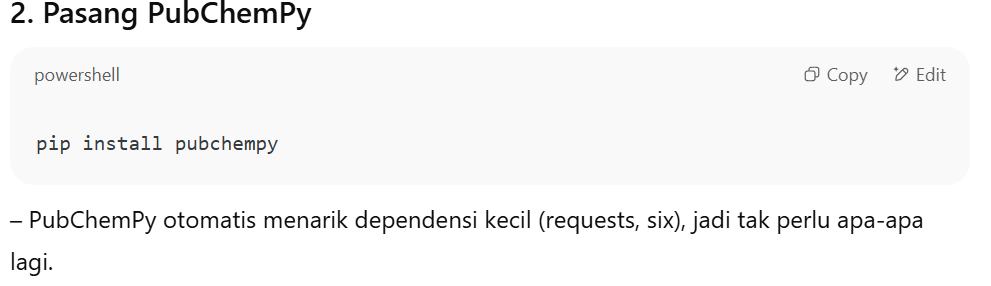
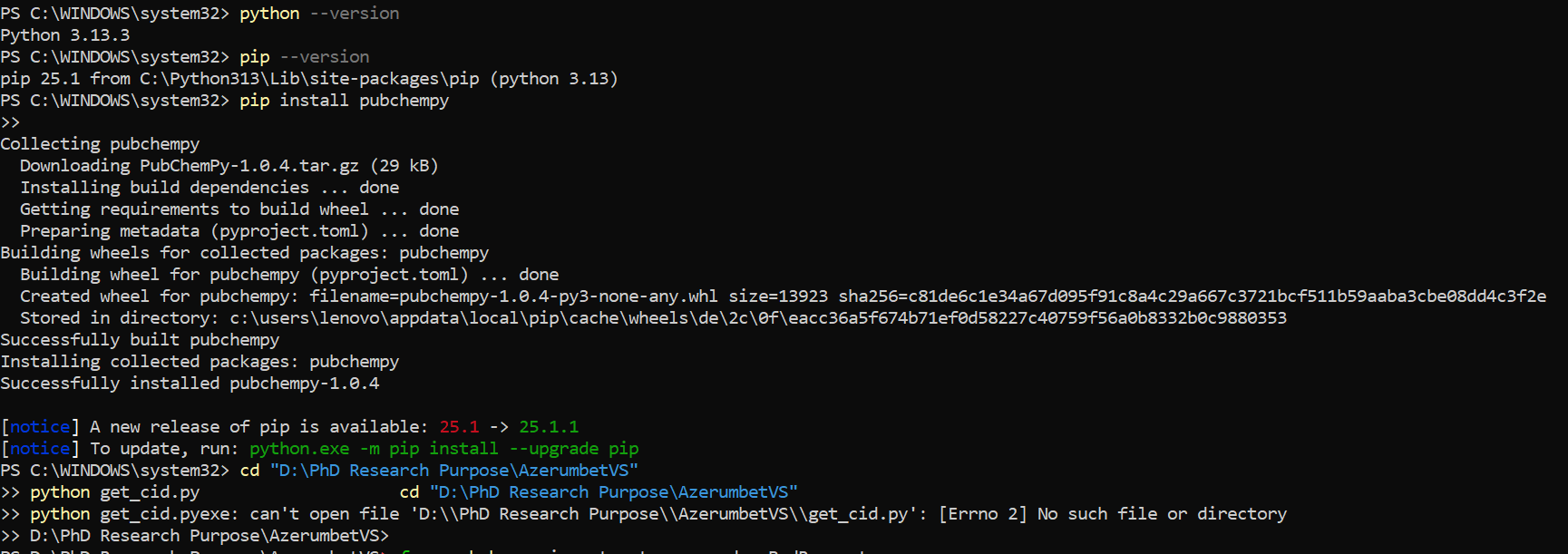
Method for Plants Virtual Screening

1. Data from [**https://doi.org/10.3390/molecules29122845**2](https://doi.org/10.3390/molecules291228452). Dibuat folder dan file simpan nama senyawa di "D:\PhD Research Purpose\AzerumbetVS\alpi\_zer\_raw.txt"
2. Dapatkan CID PubChem







1. Buat file get\_cid.py denga isi code berikut dan Run python get\_cid.py

from pubchempy import get\_compounds, PubChemHTTPError

import csv, os

RAW\_FILE = r"D:\PhD Research Purpose\AzerumbetVS\alpi\_zer\_raw.txt"

OUT\_CSV = os.path.join(os.path.dirname(RAW\_FILE), "az\_cid.csv")

with open(RAW\_FILE, encoding="utf-8") as f:

names = [l.strip() for l in f if l.strip()]

with open(OUT\_CSV, "w", newline="", encoding="utf-8") as f:

w = csv.writer(f)

w.writerow(["Name", "CID"])

for n in names:

try:

cid = get\_compounds(n, "name")[0].cid

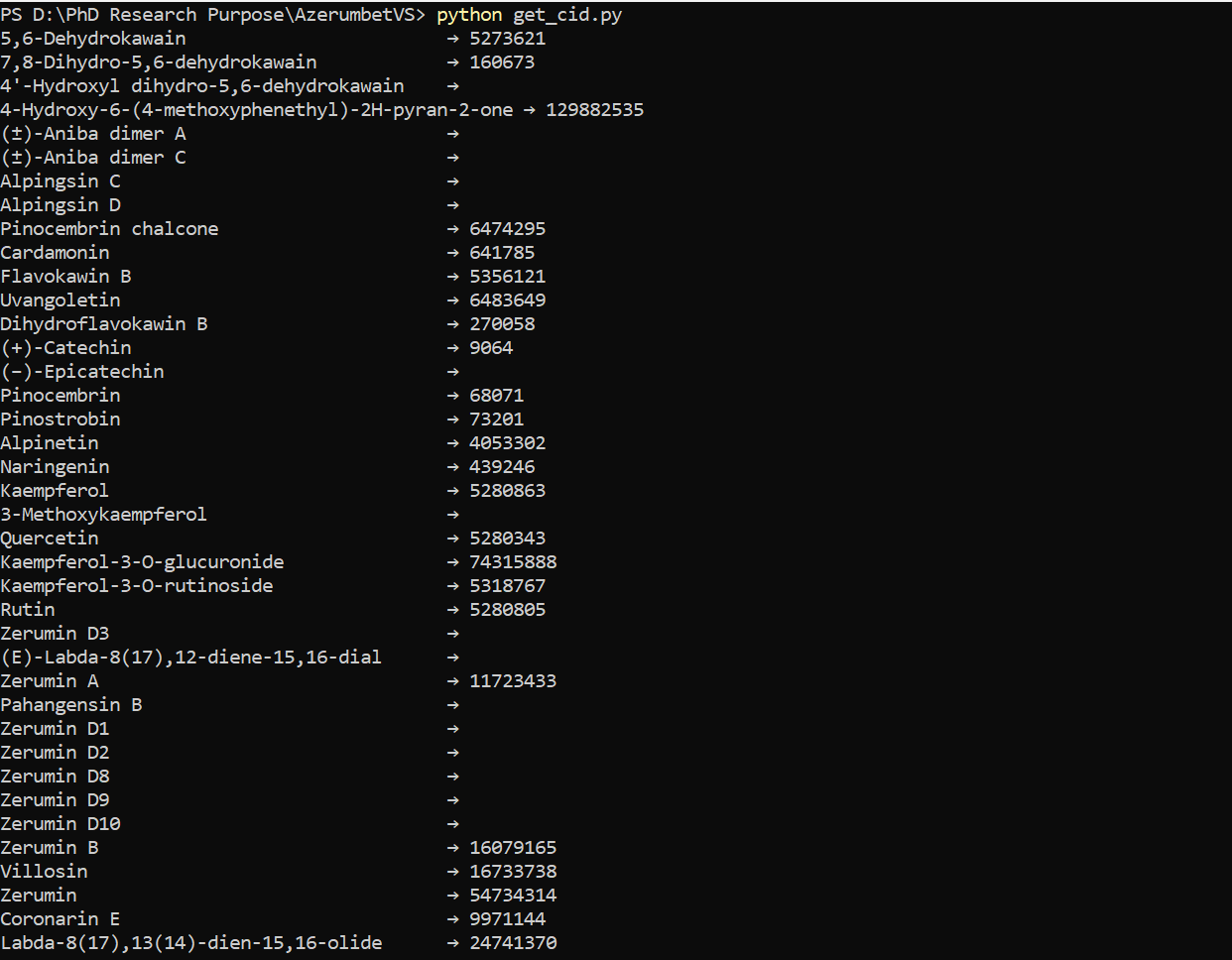
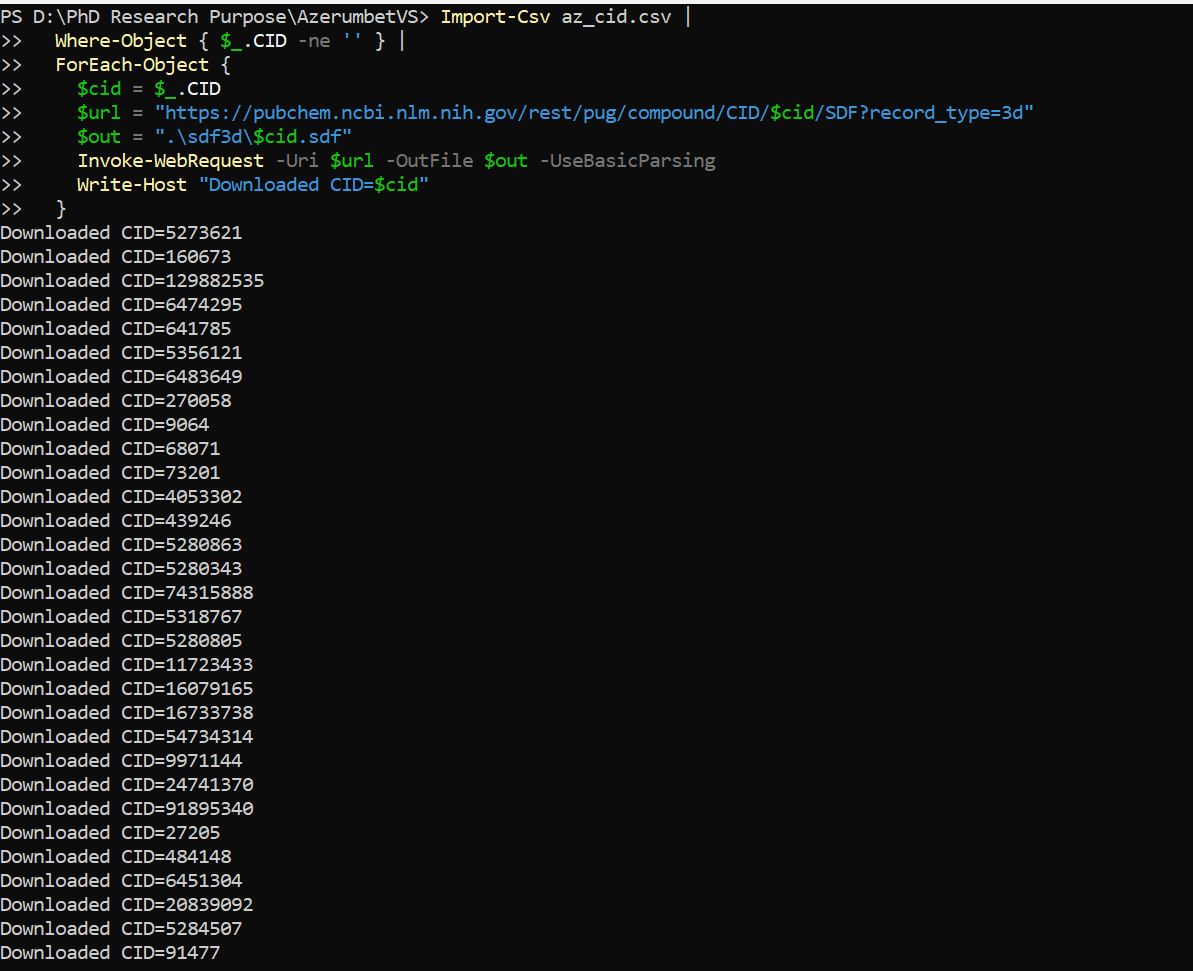
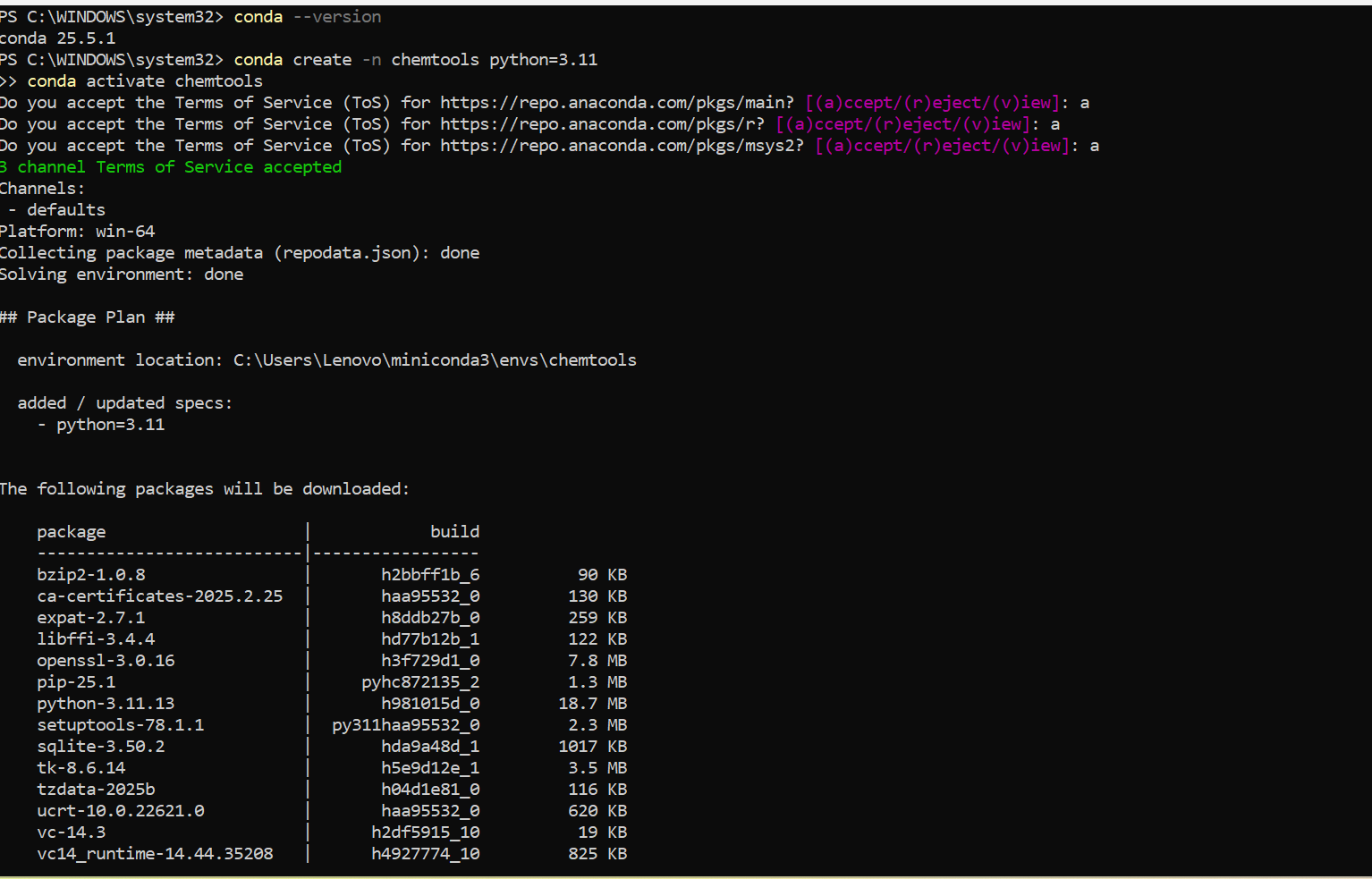
except (IndexError, PubChemHTTPError):

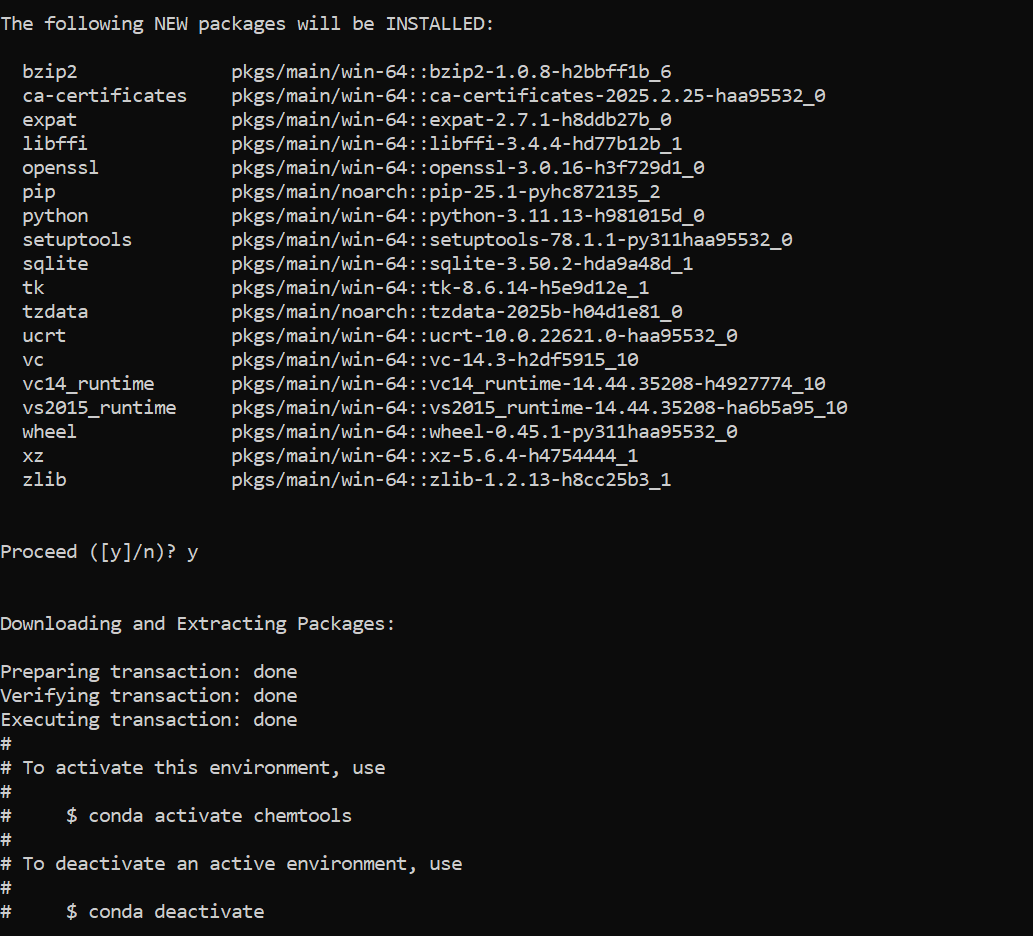
cid = ""

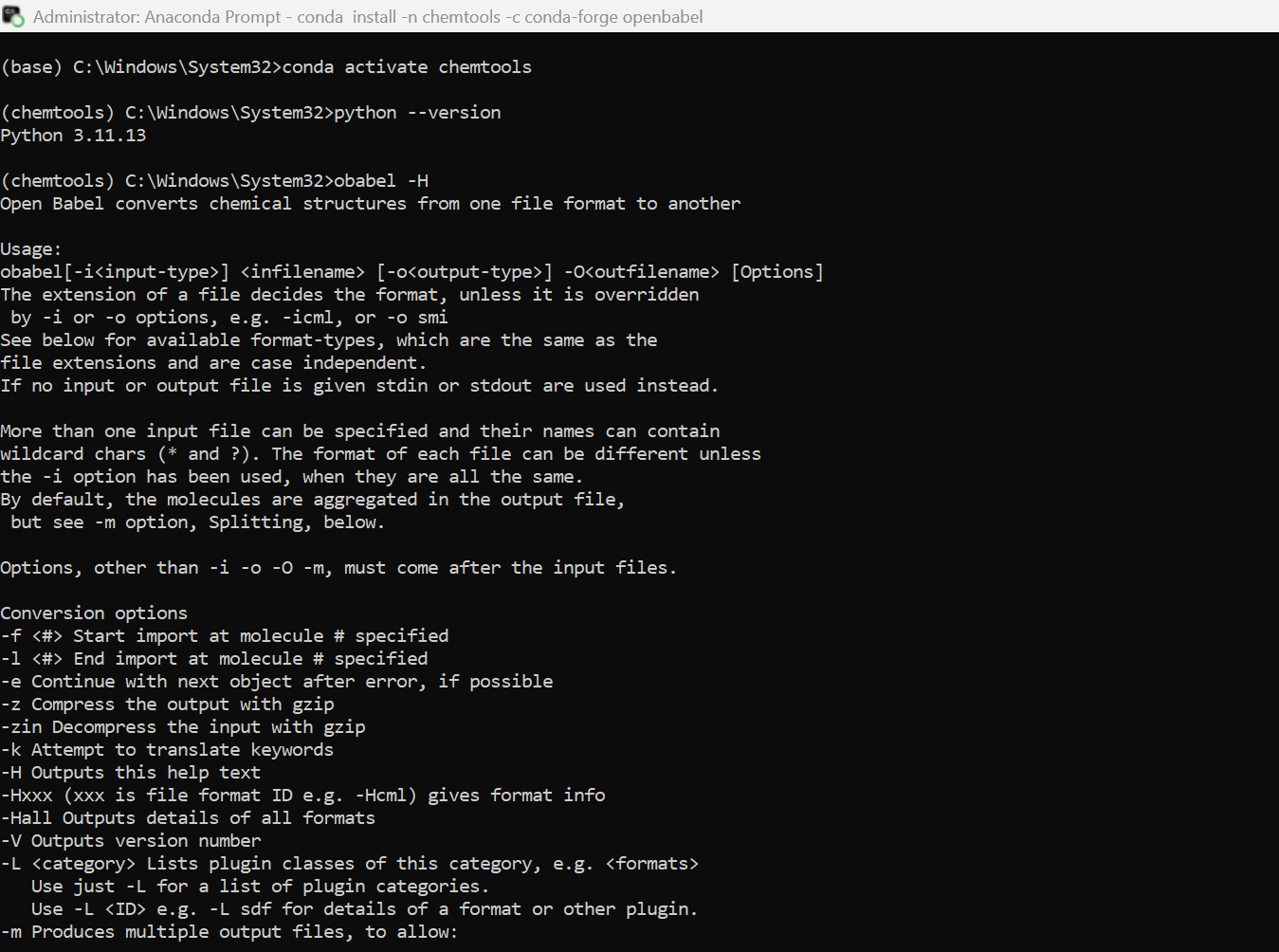
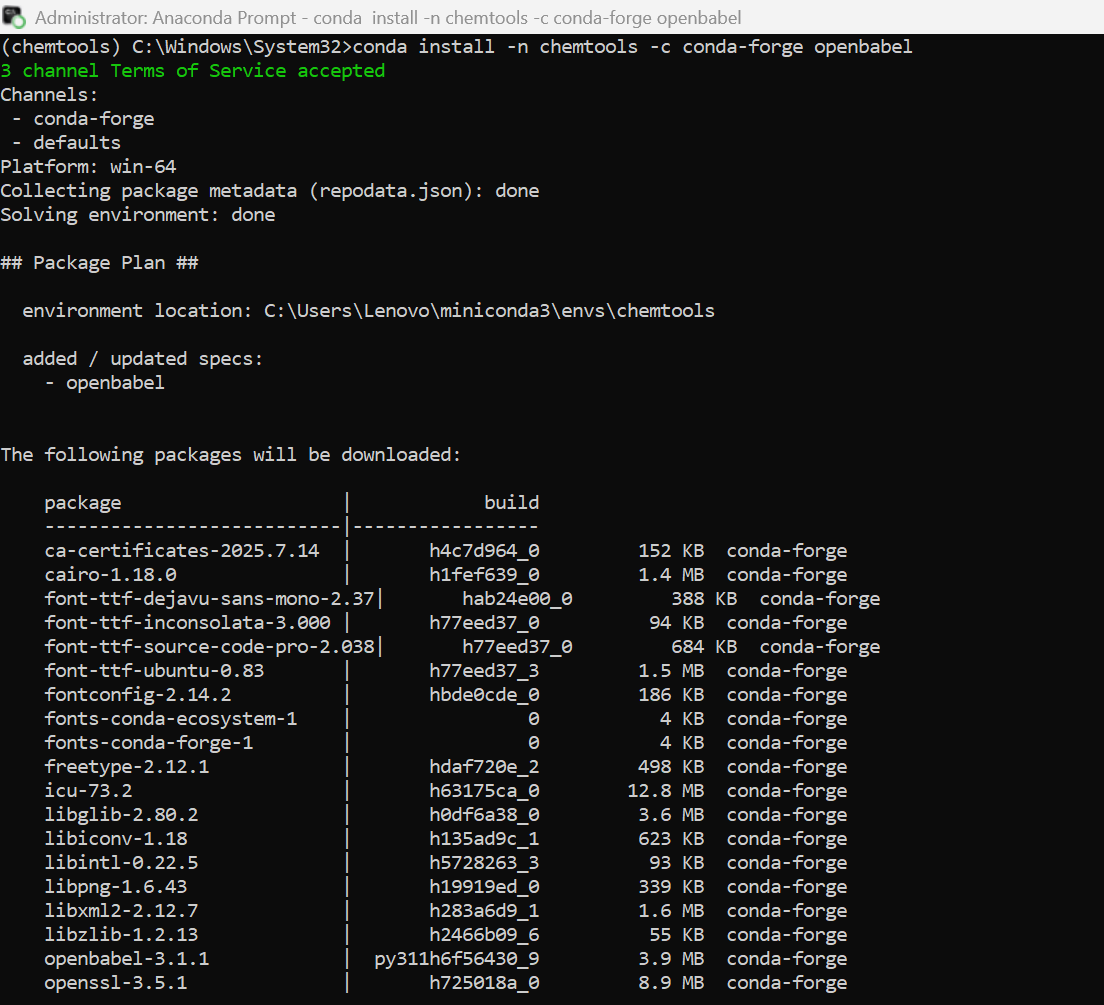
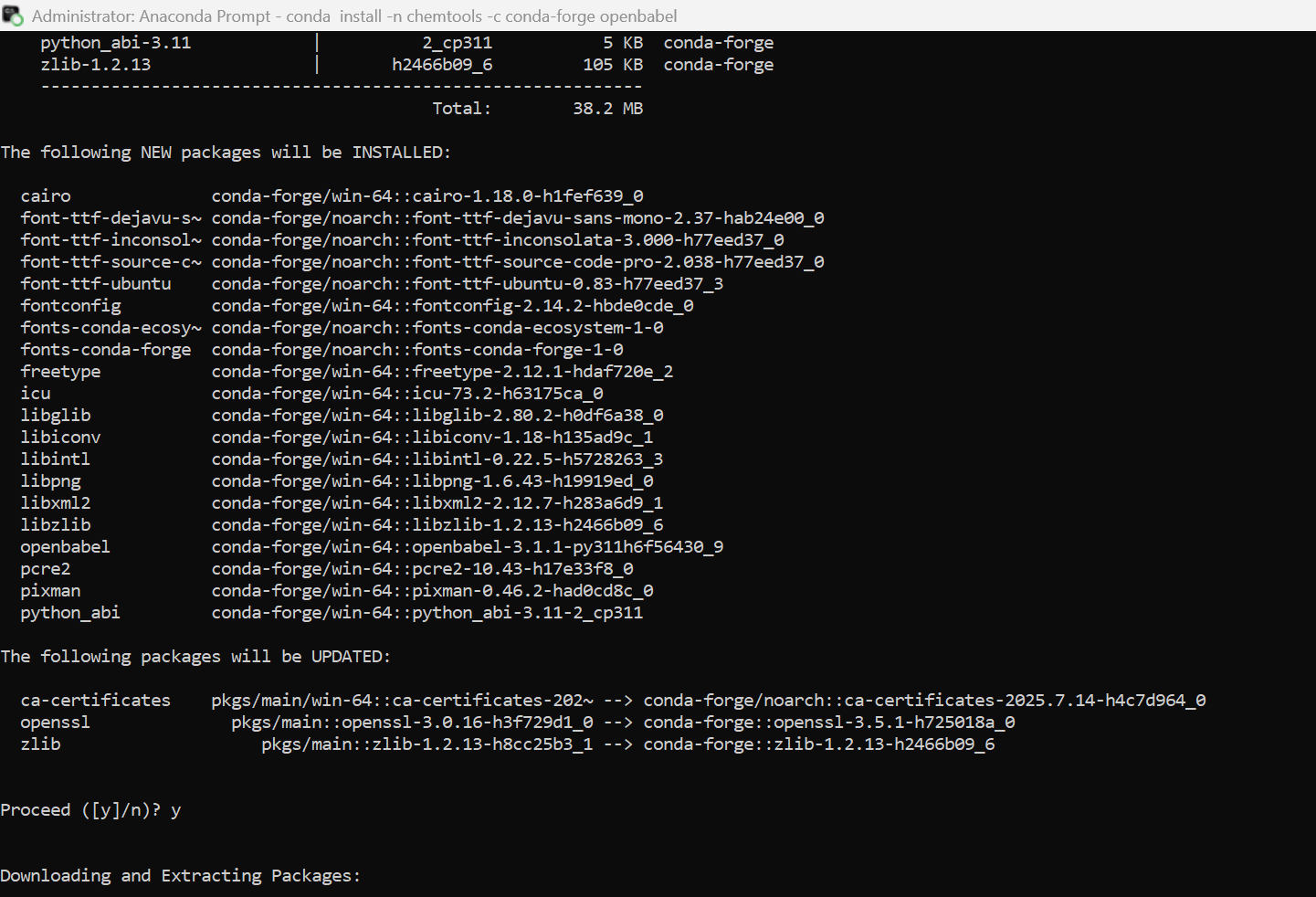
w.writerow([n, cid])

print(f"{n:40s} → {cid}")

print(f"\n✅ Selesai! Hasil tersimpan di: {OUT\_CSV}")

1. hasil  
   
2. 
3. Download conda <https://docs.conda.io/en/latest/miniconda.html> dan versi kemudian buat environment  
   



1. In Explorer, go to

makefile

CopyEdit

C:\Users\Lenovo\miniconda3\envs\chemtools\Library\share\openbabel\3.1.0\

1. **Create** a new folder called

nginx

CopyEdit

forcefields

1. From that same 3.1.0 folder, **cut & paste** all MMFF‑related files into forcefields/. In particular look for files named like:

CopyEdit

mmff94s.ff mmff94s.par mmff94.ff mmffdef.par mmffbond.par …

1. **Rename** the primary MMFF94 parameter file inside forcefields/ to **MMFF94.prm**. For example:

CopyEdit

forcefields\mmff94s.ff → forcefields\MMFF94.prm

1. Also move or rename the UFF file:

CopyEdit

forcefields\UFF.prm

(If it’s named UFF.prm already, you’re good; otherwise rename whatever UFF file you see to UFF.prm.)

After this you should have:

CopyEdit

…\3.1.0\forcefields\MMFF94.prm

…\3.1.0\forcefields\UFF.prm

1. **Rename the ring‑fragment index**
2. In the same 3.1.0 folder, you likely see a file called ring-fragments (no extension).
3. **Rename** it to exactly:

pgsql

CopyEdit

ring‑fragments‑index.txt

1. (Also ensure there’s a rigid‑fragments‑index if you see that, but the fatal error was only about the ring index.)
2. Once you’ve done those two things, verify from the prompt:

bat

CopyEdit

dir "%CONDA\_PREFIX%\Library\share\openbabel\3.1.0\forcefields\MMFF94.prm"

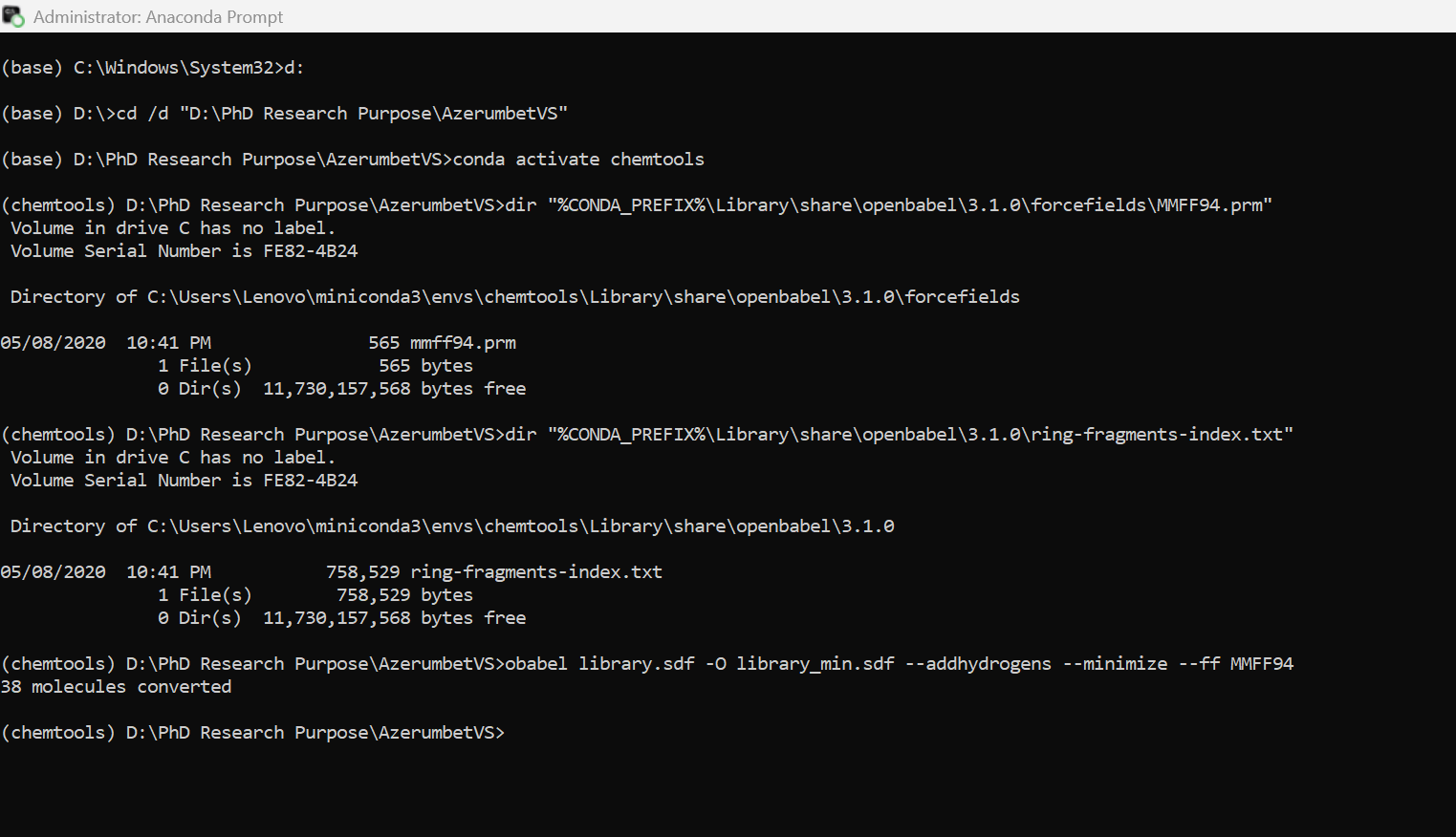
dir "%CONDA\_PREFIX%\Library\share\openbabel\3.1.0\ring-fragments-index.txt"

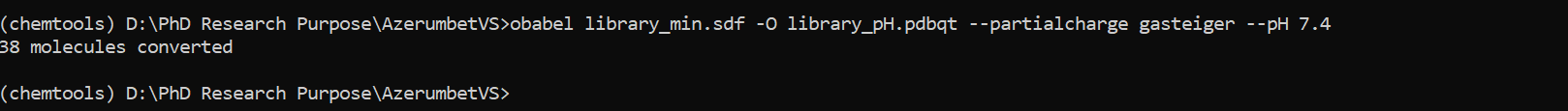
If both commands now find a file, you’re all set. Finally re‑run:

bat

CopyEdit

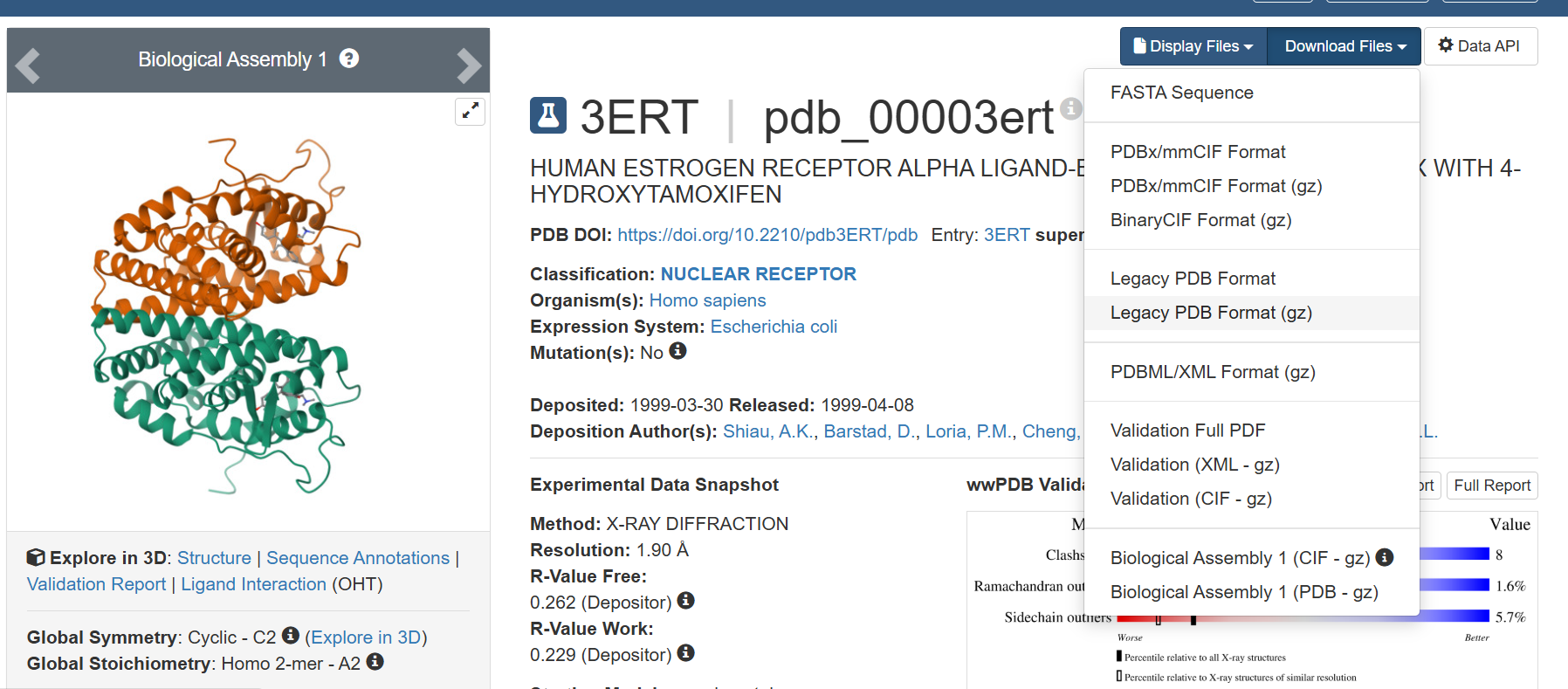
conda activate chemtools

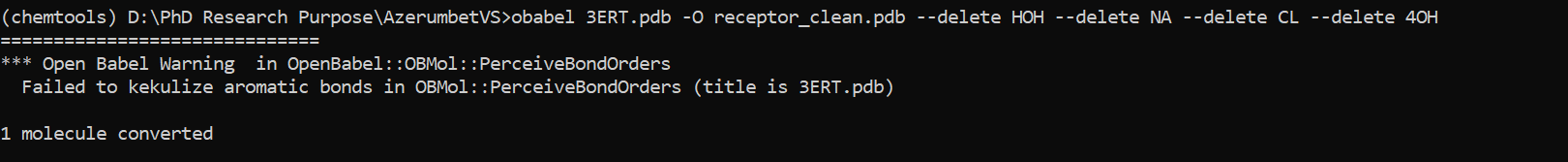
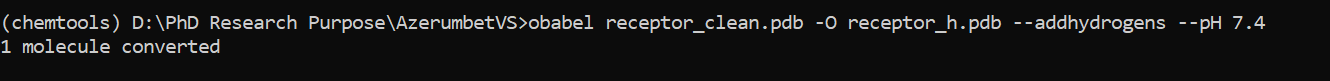
obabel library.sdf -O library\_min.sdf --addhydrogens --minimize --ff MMFF94  


1. Konversi ke PDBQT (untuk AutoDock Vina)  
   obabel library\_min.sdf -O library\_pH.pdbqt --partialcharge gasteiger --pH 7.4  
     
   
2. **Unduh Struktur ERα 3D dari PDB**

Buka web RCSB PDB: <https://www.rcsb.org>

Cari “Estrogen Receptor alpha ligand binding domain” atau langsung masukkan PDB ID, misalnya **3ERT** (ERα bound to 4-hydroxytamoxifen).

Download file **3ERT.pdb**.  


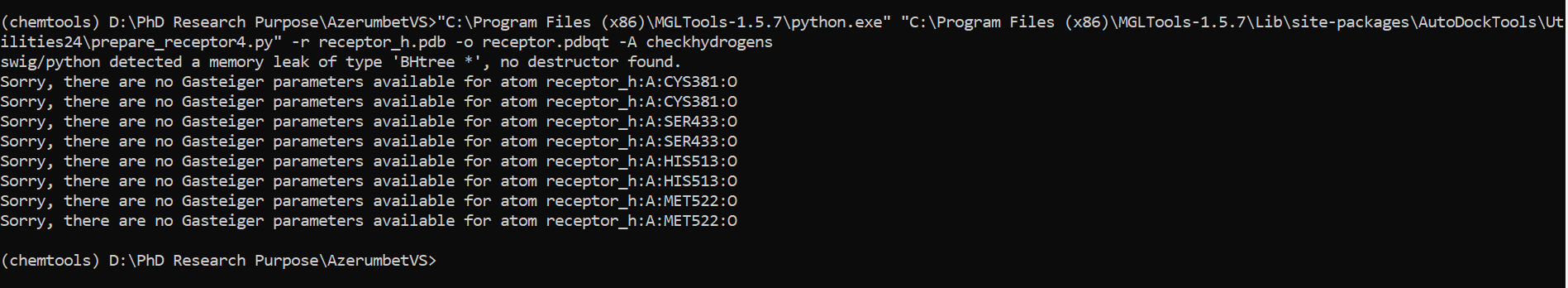
1. obabel 3ERT.pdb -O receptor\_clean.pdb --delete HOH --delete NA --delete CL --delete 4OH  
     
   
2. obabel receptor\_clean.pdb -O receptor\_h.pdb --addhydrogens --pH 7.4  
   
3. Buka halaman unduh MGLTools:

http://mgltools.scripps.edu/downloads

Pilih installer yang sesuai Windows (biasanya MGLTools-1.5.7-win32.exe).

Jalankan installer, ikuti instruksi, dan centang opsi “Add MGLTools to your PATH” jika tersedia.

Setelah selesai, tutup dan buka ulang Anaconda Prompt agar prepare\_receptor4.py dikenali.  
**Generate receptor.pdbqt**

Dengan asumsi kamu sudah punya receptor\_h.pdb (protein bersih + hydrogen), jalankan:  
"C:\Program Files (x86)\MGLTools-1.5.7\python.exe" "C:\Program Files (x86)\MGLTools-1.5.7\Lib\site-packages\AutoDockTools\Utilities24\prepare\_receptor4.py" -r receptor\_h.pdb -o receptor.pdbqt -A checkhydrogens  


Penjelasan opsi:

* -r receptor\_h.pdb → input file PDB yang sudah ditambahkan hydrogen
* -o receptor.pdbqt → output file PDBQT siap docking
* -A checkhydrogens → perintahkan script memeriksa & memperbaiki atom H duplikat atau hilang

Outputnya adalah **receptor.pdbqt**, di mana:

* Tiap atom sudah diberi tipe (C, N, O, H…) sesuai AutoDock
* Muatan Kollman telah dihitung dan tertulis
* Struktur sudah siap dipakai oleh AutoDock Vina

AutoDock Vina tidak “mencari” di seluruh ruang 3D secara global—ia hanya mengeksplor dalam sebuah **kotak (grid box)** yang Anda tentukan dengan --center\_x/--center\_y/--center\_z dan --size\_x/--size\_y/--size\_z

1.  **Fokus pada situs aktif**  
   Anda sudah tahu situs aktif (binding pocket) dari struktur co‑crystal. Dengan memberi tahu Vina “mulai cari di sini,” docking jadi jauh lebih cepat dan akurat.

 **Hemat waktu komputasi**  
Ruang pencarian tersempit → lebih sedikit posisi yang harus dihitung.

1. **Buat file** centroid.py di folder projek, isinya:

# centroid.py (updated)

import sys

pdbfile = sys.argv[1]

resn = sys.argv[2]

coords = []

with open(pdbfile) as f:

for L in f:

if (L.startswith("HETATM") or L.startswith("ATOM ")) and L[17:20] == resn:

x = float(L[30:38]); y = float(L[38:46]); z = float(L[46:54])

coords.append((x,y,z))

if not coords:

print("Tidak menemukan ligand", resn)

sys.exit(1)

cx = sum(p[0] for p in coords) / len(coords)

cy = sum(p[1] for p in coords) / len(coords)

cz = sum(p[2] for p in coords) / len(coords)

print(f"centroid: {cx:.3f} {cy:.3f} {cz:.3f}")

1. Temukan kode ligand di 3ERT.pdb

Di Anaconda Prompt, jalankan:

findstr /B HETATM 3ERT.pdb > hetatm.txt

Lalu buka hetatm.txt dengan Notepad. Di kolom ke‑4 (huruf 18–21) kamu akan lihat tiga‑huruf kode untuk setiap HETATM. Cari baris yang bukan air (“HOH”) atau ion, misalnya akan ada:

HETATM 1361 C1 XXX A 401 12.345 -3.212 25.784 1.00 0.00 ...

Di situ XXX adalah kode ligand co‑crystal-mu.

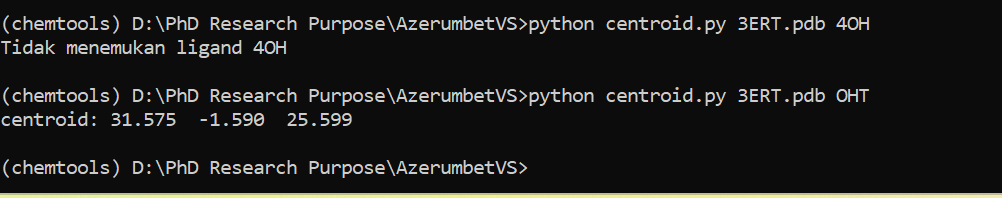
Gunakan kode ligand yang tepat

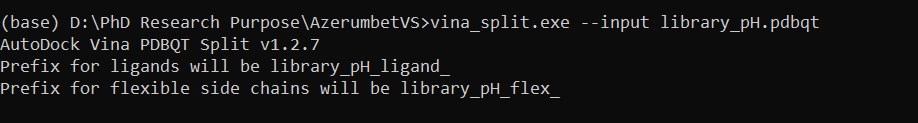
Misalnya kamu melihat ligand tercantum sebagai “TOT” (atau “4OH” mungkin sebenarnya “OHT”, dsb.), maka jalankan ulang skrip:

*python centroid.py 3ERT.pdb TOT atau OHT atau 4OH*

—ganti TOT dengan kode yang kamu temukan.

Jika sudah berhasil, dia akan mencetak:



1. Untuk docking si receptor dari ligand uji coba dgn prompt berikut :  
   *vina\_split.exe --input library\_pH.pdbqt*  
   

**Pastikan** vina.exe sudah ada di %PATH% (anda sudah cek where vina.exe → C:\Tools\AutoDock‑Vina\vina.exe).

**Mulai satu perintah**:

vina ^

--receptor receptor.pdbqt ^

--ligand library\_pH\_ligand\_01.pdbqt ^

--out docking\_out\_01.pdbqt ^

--center\_x 31.575 ^

--center\_y -1.590 ^

--center\_z 25.599 ^

--size\_x 20 ^

--size\_y 20 ^

--size\_z 20 ^

--cpu 4

* + Simbol caret (^) **harus** jadi karakter terakhir di baris, tanpa spasi setelahnya.
  + Anda tidak boleh menekan Enter **tanpa** meletakkan caret di akhir baris, atau Windows akan menjalankan baris itu sebagai perintah tersendiri.

Atau, kalau susah di cmd, pindahkan seluruh blok itu ke dalam sebuah file **batch** (misalnya dock01.bat):

@echo off

vina ^

--receptor receptor.pdbqt ^

--ligand library\_pH\_ligand\_01.pdbqt ^

--out docking\_out\_01.pdbqt ^

--center\_x 31.575 ^

--center\_y -1.590 ^

--center\_z 25.599 ^

--size\_x 20 ^

--size\_y 20 ^

--size\_z 20 ^

--cpu 4

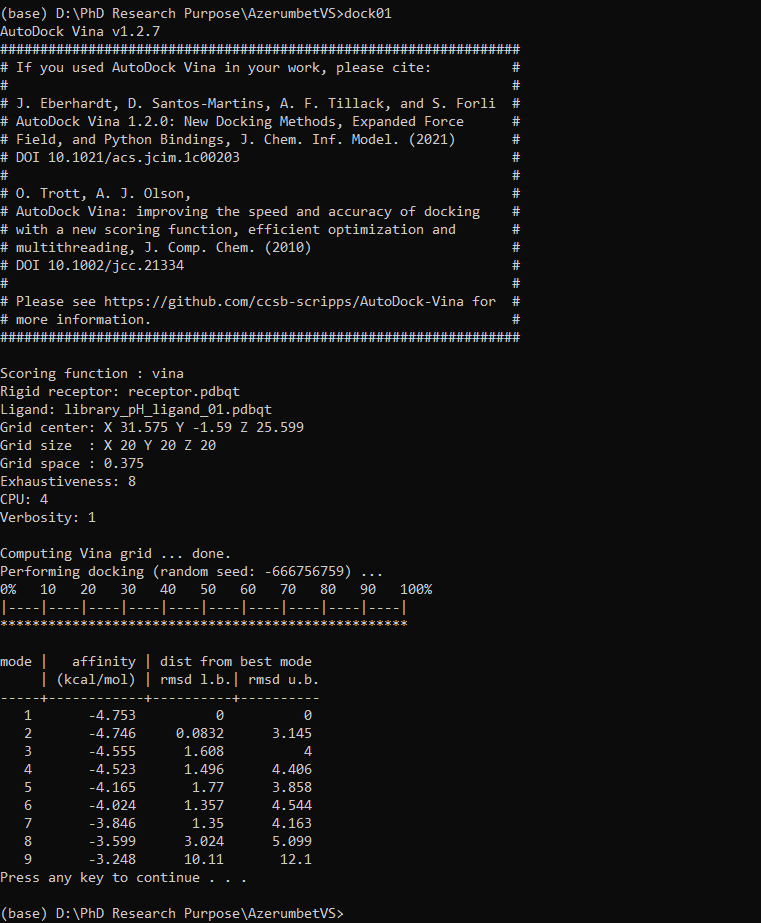
pause

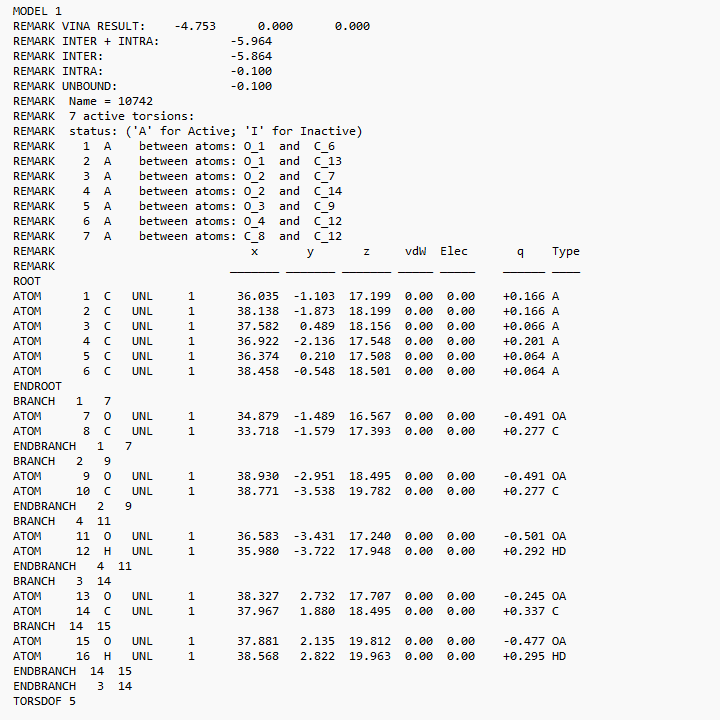
Lalu cukup jalankan dock01.bat.

**Ringkasnya**

* **Jangan** mengetik --center\_x dll. secara terpisah—mereka *options* yang harus menyatu dengan vina.exe.
* Gunakan caret (^) **di akhir** setiap baris untuk *line continuation* di Windows cmd.
* Alternatifnya: bungkus jadi satu file .bat agar mudah dijalankan ulang atau di‑loop untuk semua ligand.

Dengan ini, Vina akan mengerti bahwa semua baris tersebut adalah bagian dari satu invokasi program yang sama dan akan benar‑benar melakukan docking, bukan sekadar mencetak parameter.





**Cara membaca ringkas**

**Ambil skor VINA\_RESULT** paling pertama (mode 1), misalnya −4.753 kcal/mol → itu prediksi afinitas terbaik.

**Cek INTER vs INTRA** jika kamu tertarik seberapa besar penalti internal ligand.

**Tampilkan pose** di visualizer (PyMOL, Chimera, dsb.) menggunakan file docking\_out\_01.pdbqt untuk melihat orientasi ligand di dalam kotak (grid).

**Bandingkan** skor antar pose (mode 1, 2, 3…) jika kamu generate lebih banyak (--num\_modes).

Itu saja — di docking sederhana dengan Vina biasanya kita ambil mode 1 saja karena memiliki energi paling rendah (paling negatif), lalu lihat interaksinya (ikatan hidrogen, tumpang tindih van der Waals, dsb.) di software visualisasi.

1. Berikut langkah‐langkah dasar untuk memuat hasil docking (receptor + pose Vina) ke PyMOL:
2. **Persiapkan file**

receptor.pdbqt → ubah ekstensi jadi .pdb, kita akan merender hanya koordinat atom (PyMOL tidak langsung baca PDBQT)

docking\_out.pdbqt → berisi pose Vina (MODEL 1, MODEL 2, …)

Kita akan mengekstrak MODEL 1 dari docking\_out.pdbqt ke file terpisah:

*# ambil MODEL 1 saja*

*vina\_split.exe --input docking\_out.pdbqt*

*#hasilnya:*

*# docking\_out\_ligand\_1.pdbqt ← MODEL 1*

*# docking\_out\_flex\_1.pdbqt ← (kosong, karena tak ada flex-residue)*

Rename:

*ren docking\_out\_ligand\_1.pdbqt pose1.pdbqt*atau kalo langsung semua file ligand-PH nya berikut code di file .bat nya  
*@echo off*

*cd /d "D:\PhD Research Purpose\AzerumbetVS"*

*REM — Buat folder jika belum ada*

*if not exist docking\_results (*

*mkdir docking\_results*

*)*

*REM — Jalankan batch docking*

*vina ^*

*--receptor receptor.pdbqt ^*

*--batch "ligand to docking" ^*

*--dir docking\_results ^*

*--center\_x 31.575 ^*

*--center\_y -1.590 ^*

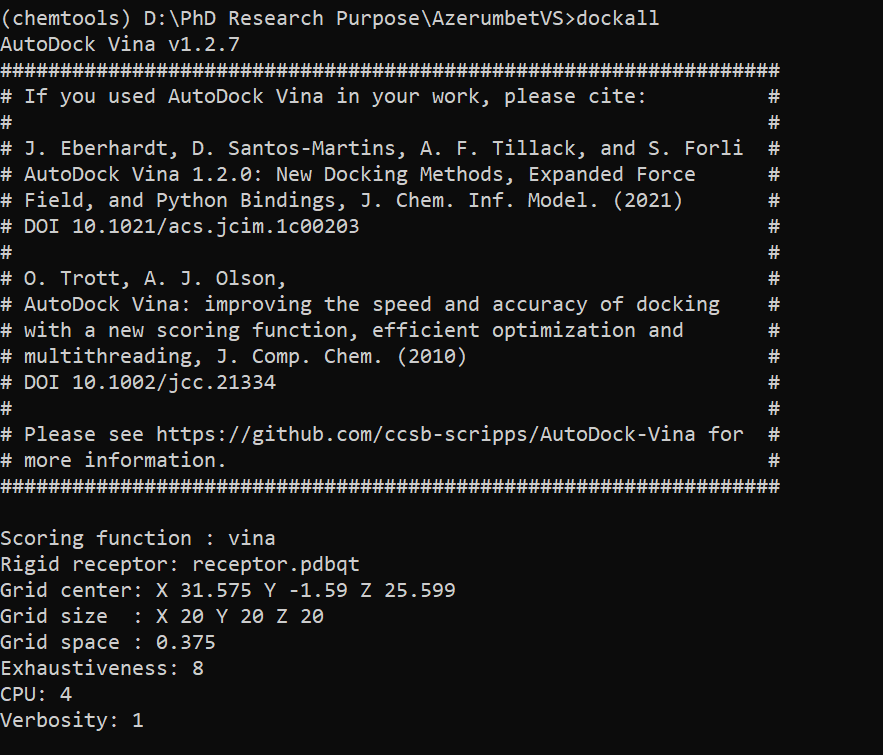
*--center\_z 25.599 ^*

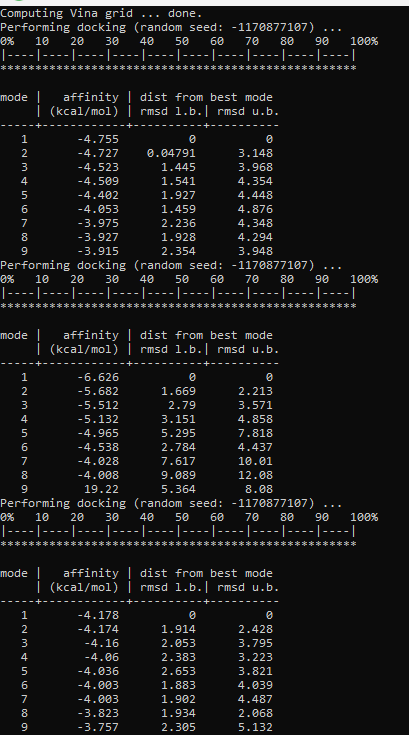
*--size\_x 20 ^*

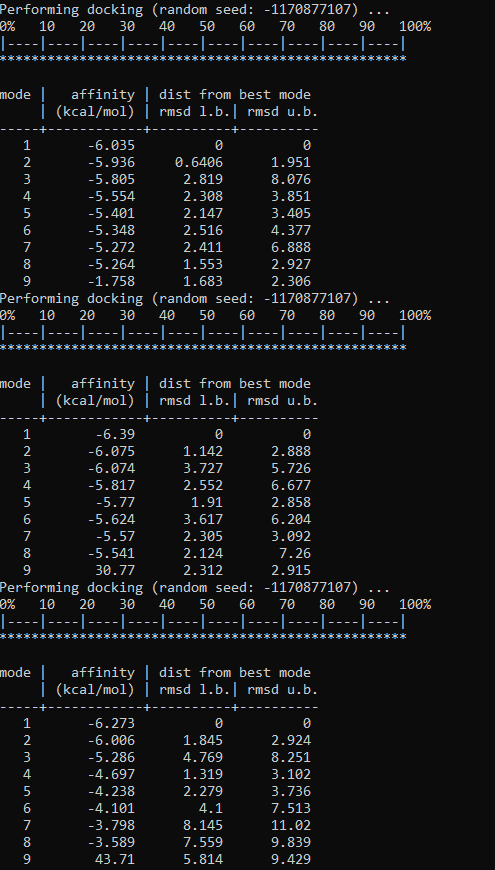
*--size\_y 20 ^*

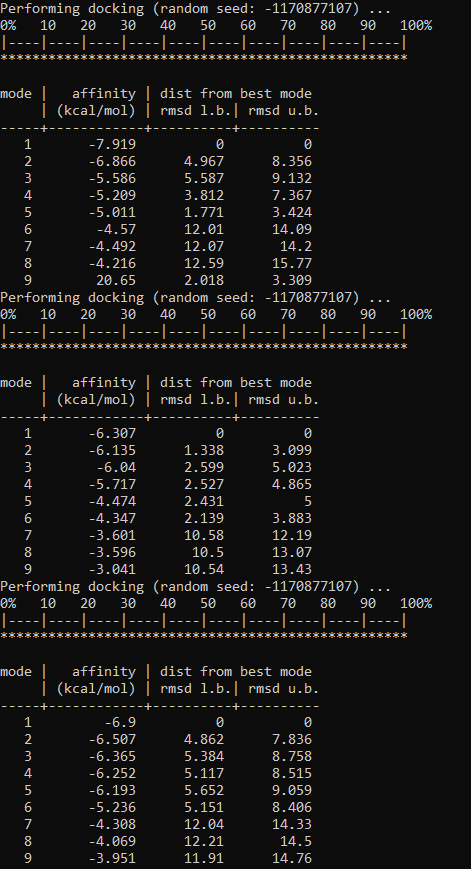
*--size\_z 20 ^*

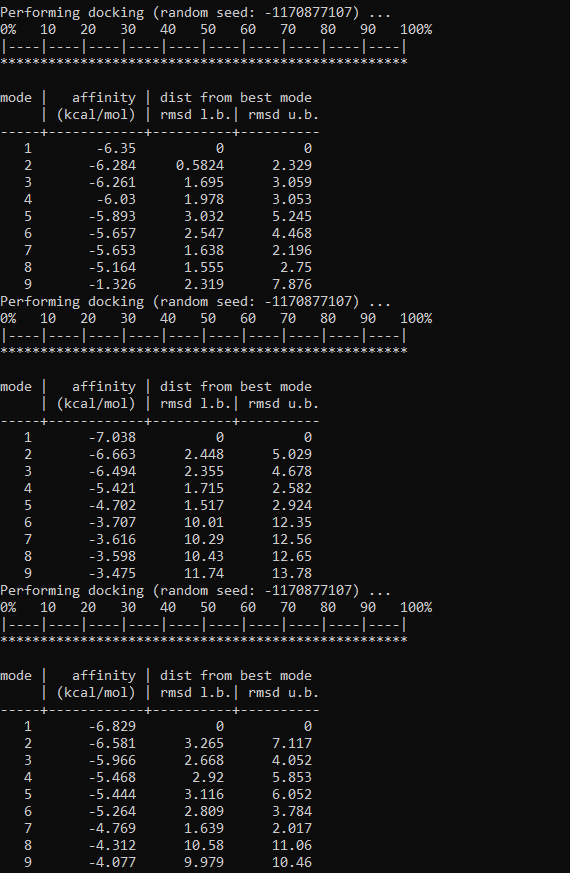
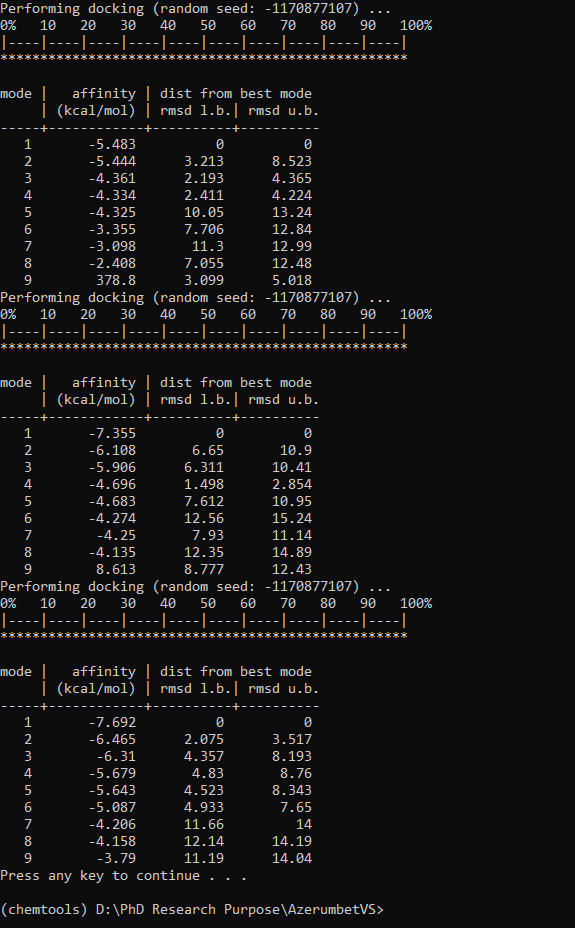
*--cpu 4*

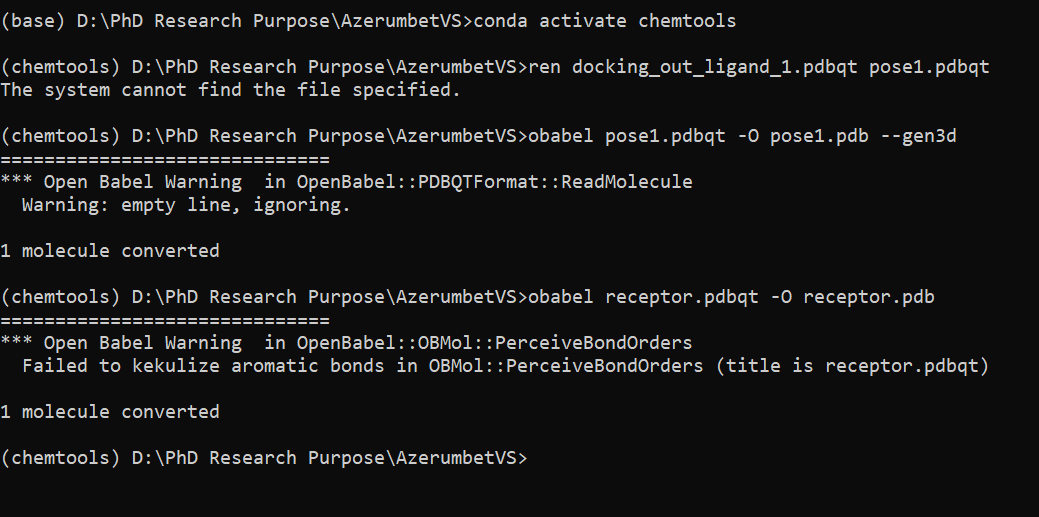
*Pause*







1. 
2. 