

# Molecular Docking with Chimera and AutoDockVina

## Part II

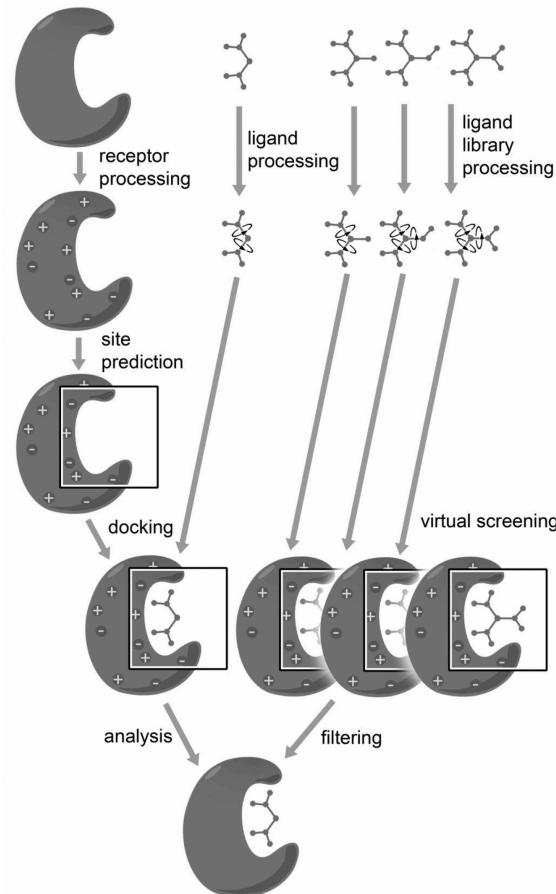
---

**CCATS Group**



# The History of AutoDock [1]

1. **Review:** Goodsell, Sanner, Olson, and Forli, Protein Sci (2021). doi: 10.1002/pro.3934
2. **AutoDock:** Goodsell and Olson, Proteins (1990). doi: 10.1002/prot.340080302
3. **AutoDock 4.2:** Morris, Huey, and Lindstrom, Sanner, Belew, Goodsell, and Olson, J Comp Chem (2009). doi: 10.1002/jcc.21256
4. **AutoDock Vina:** Trott and Olson, J Comp Chem (2010). doi: 10.1002/jcc.21334
5. **AutoDockFR:** Ravindranath, Forli, Goodsell, Olson, and Sanner, PLoS Comput Biol (2015). doi: 10.1371/journal.pcbi.1004586
6. **AutoDockCrankRep (peptide docking):** Zhang and Sanner, Bioinformatics (2019). doi: 10.1093/bioinformatics/btz459

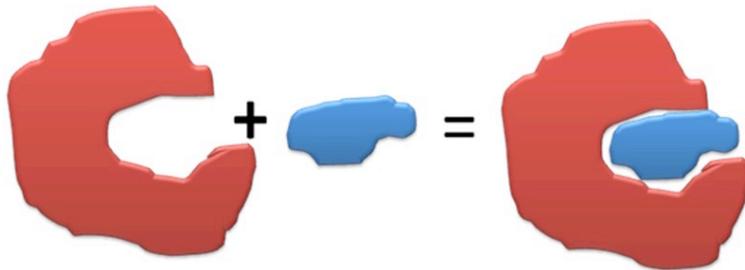


# Choosing a Docking Engine [1]

	AutoDock4	AutoDockGPU	AutoDock Vina	ADFR	ADCP
Rapid docking of drug-like molecules			✓		
Very flexible ligands? (>20 torsions)		✓	✓	✓	
Flexible macrocycle	✓			✓	
Is the ligand a peptide?					✓
Are waters important for binding?	✓	✓		✓	
Is the binding site flexible?			✓	✓	
Is there a metal ion in the site?	✓	✓			✓
Covalent inhibitor (known site)?	✓	✓	✓		✓
Covalent inhibitor (unknown site)?	✓				
Anchored docking				✓	

## The Scoring Function

---



- [7] Huey, Morris, Olson and Goodsell, J Comput Chem (2007). doi: 10.1002/jcc.20634
- The AutoDock force field is [7]

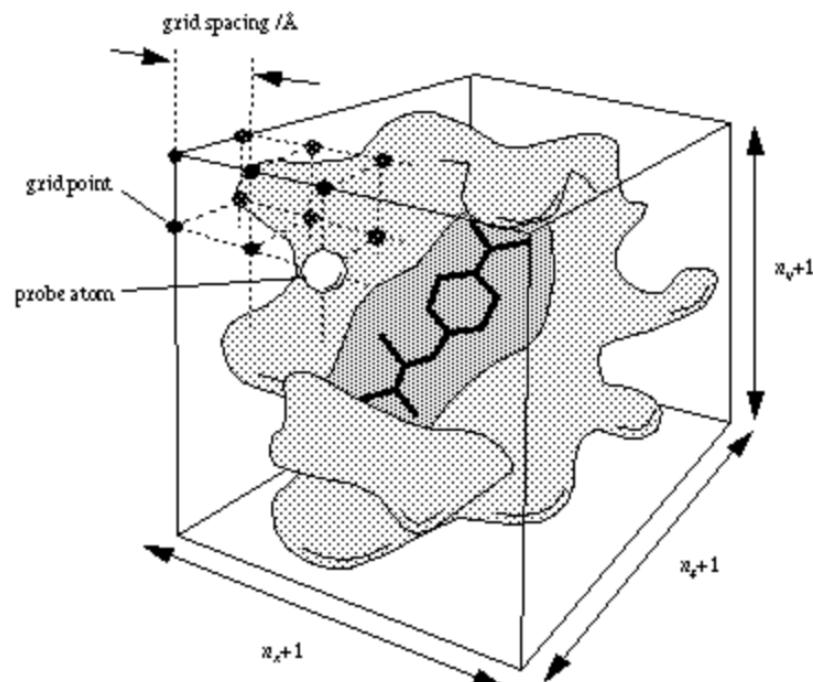
$$V = W_{vdw} \sum_{i,j} \left[ \frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^6} \right] + W_{hb} \sum_{i,j} E(t) \left[ \frac{C_{ij}}{r_{ij}^{12}} - \frac{D_{ij}}{r_{ij}^{10}} \right] + W_{elec} \sum_{i,j} \frac{q_i q_j}{\epsilon(r_{ij}) r_{ij}} + W_{sol} \sum_{i,j} (S_i V_j + V_i S_j) \exp\left(-\frac{r_{ij}^2}{2\sigma^2}\right)$$

- The free energy scoring function is

$$\Delta G = (V_{bound}^{L-L} - V_{unbound}^{L-L}) + (V_{bound}^{P-P} - V_{unbound}^{P-P}) + (V_{bound}^{P-L} - V_{unbound}^{P-L} + \Delta S_{conf})$$

## The Search Algorithm

- Needs to explore enough of the available conformational space in a timely manner
- For a protein target, **volumetric maps** are precalculated [3]
  - the target protein is embedded in a grid
  - a probe atom is sequentially placed at each grid point
  - the interaction energy between the probe atom and the target is computed
  - the value for the interaction energy is saved on the grid
  - See also [http://www.csb.yale.edu/userguides/datamanip/autodock/html/Using\\_AutoDock\\_305.1.html](http://www.csb.yale.edu/userguides/datamanip/autodock/html/Using_AutoDock_305.1.html)
- AutoDock 4 Search is carried out by combining Lamarckian genetic algorithm and local search
- AutoDock Vina search is carried out by combining Monte Carlo algorithm and local search



# Virtual Screening

---

- Automated evaluation of compound libraries for synthesis, purchase, or testing
- Predicting the binding affinity of Venetoclax to Bcl-2 ~1 min
- Does this mean we can only screen ~1,400 compounds/day ????
  - Maximize the use of computational resources, typically performed on a high performance cluster
- The general protocol is:
  - Compound library preparation
  - Protein preparation
  - Your favorite docking program

## Compound Libraries

---

- NCI Diversity Sets (<https://wiki.nci.nih.gov/display/ncidtpdata/compound+sets>)
- Zinc Database (<https://zinc.docking.org>)
- These files should contain 3D coordinates for the compound of interest
- For virtual screening with AutoDock Vina, the compound files must be PDBQT format
- Custom libraries
  - ChemDraw
  - RDkit
  - Obabel

# Tutorial

---

- **Goal:** Virtually screen Bcl-2 (PDB 6O0K) to identify new compounds for the BH3-binding domain
- Compound library contains the NCI Diversity Set 5 (99/1593 compounds), and Venetoclax
- Docking with AutoDock Vina:
  - Single compound
  - Virtual Screening
    - Without using external libraries
    - With GNU Parallel (1, 2, 4, or 8 cores/job)
- PDBQT files can be made with AutoDock Tools (<https://ccsb.scripps.edu/mgltools/downloads/>)
- In this tutorial, the \*.receptor.pdbqt and search box information was obtained from last weeks Bcl-2 docking. The Diversity Set 5 compounds was converted with OBabel.

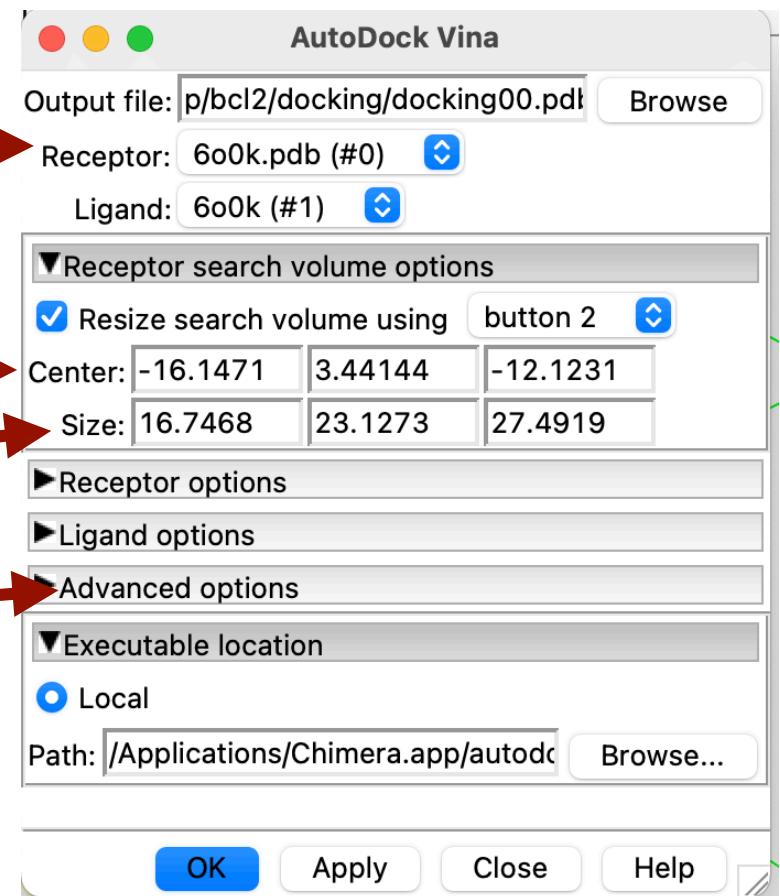
## dock.conf Example

```
(base) [van@schooner2 gnu_parallel_2core]$ cat dock.conf
# Configuration file is made by viewing the box on UCSF Chimera running AutoDock Vina.
# Exhaustiveness is dependent on box size

center_x =
center_y =
center_z =

size_x =
size_y =
size_z =

cpu=
exhaustiveness
receptor =
```



- Exhaustiveness - number of independent runs from random initial configurations
- Each run: 1) random perturbation, 2) local optimization (evaluates scoring function), and 3) selection

## Docking Venetoclax to Bcl-2 (Command Line)

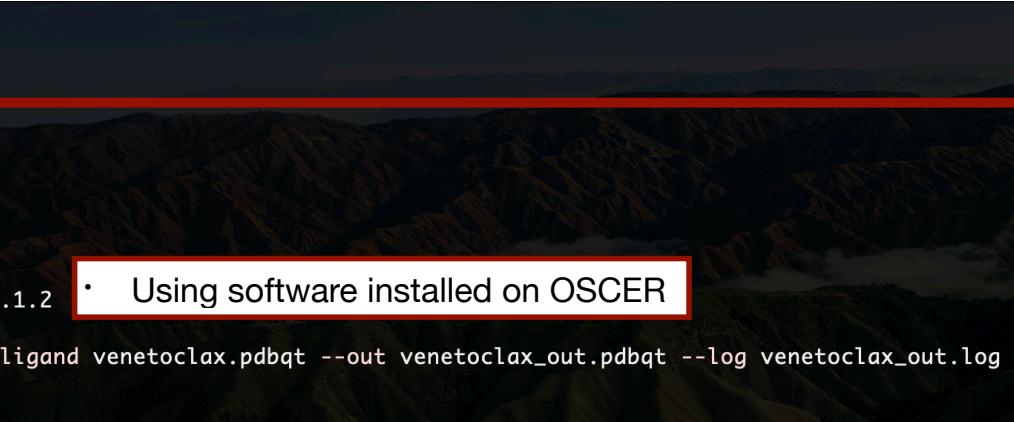
---

- Files:
  - venetoclax.pdbqt (ligand)
  - 6o0k\_protein\_clean.pdbqt (receptor)
  - dock.conf (information on protein file, search box, and CPU)
  - rundocking.slurm (SLURM submission script to run vina on HPC)
- Steps:
  - Modify the dock.conf file to include search box info from previous tutorial
  - Run job:
    - `sbatch rundocking.slurm`

## venetoclax (rundocking.slurm - the submitting script)

- First line is always the shebang

```
1 #!/bin/bash
2 #SBATCH --partition=debug
3 #SBATCH --ntasks=4
4 #SBATCH --output=%j.out
5 #SBATCH --error=%j.err
6 #SBATCH --time=00:05:00
7 #SBATCH --job-name=v
8
9 date
10 module load GCCcore/4.9.3
11 module load AutoDockVina/1.1.2
12
13 vina --config dock.conf --ligand venetoclax.pdbqt --out venetoclax_out.pdbqt --log venetoclax_out.log
14
15 date
```



• Using software installed on OSCER

- Once the modules are loaded you can call “vina”
- - - config : point to the configuration file
- - - ligand : point to the ligand file
- - - out : output the docking result
- - - log : print a log file

- SLURM batch commands for submitting jobs
  - Partition - group of nodes based on hardware (debug is for short 30 min jobs)
  - ntask - # of tasks expected
  - Output/error - gives output/error file with job ID as name
  - Time - requested time

# virtual screening

---

- The folder is organized as follows:
  - Docking (template files)
    - no\_external\_lib (Made for readability)
    - dock.conf (information on protein file, search box, and CPU)
    - Compounds (compound library)
    - Receptor (protein file)
    - setup.sh (makes a list of compound names; returns job\_id.index)
    - rundocking.slurm (submission script for VS of 10 compounds at a time)
  - Results (Contains my docking results and CPU test runs)
  - gnu\_parallel\_2core (Uses GNU parallel to optimize serial jobs)
    - dock.conf (information on protein file, search box, CPU)
    - Compounds (compound library)
    - Receptor (protein)
    - setup.sh (makes a list of compound names; returns job\_id.index)
    - rundocking.sh (Takes argument, finds compound, runs Vina)
    - runparallel.slurm (submission script for VS, 2 cores/job (10 jobs))

## **no\_external\_lib (~13 min)**

---

- Go to the folder
  - Cd virtual\_screening/no\_external\_lib
- Prepare the compound list by:
  - Bash setup.sh
- Modify the dock.conf for the search volume, CPU, and location of the protein file
- Submit the SLURM script:
  - Sbatch rundocking.slurm

## no\_external\_lib (rundocking.slurm - the submitting script)

```
1#!/bin/bash
2#SBATCH --partition=debug
3#SBATCH --exclusive
4#SBATCH --ntasks=20
5#SBATCH --output=%j.out
6#SBATCH --error=%j.err
7#SBATCH --time=00:30:00
8#SBATCH --job-name=vina
9
10 date
11 module load GCCcore/4.9.3
12 module load AutoDockVina/1.1.2
13
14 mkdir -p docking
15
16 cat job_id.index | while read -r line1; do
17     read -r line2
18     read -r line3
19     read -r line4
20     read -r line5
21     read -r line6
22     read -r line7
23     read -r line8
24     read -r line9
25     read -r line10
26     vina --config dock.conf --ligand compounds/${line1}.pdbqt --out docking/${line1}_out.pdbqt --log docking/${line1}_out.log &
27     vina --config dock.conf --ligand compounds/${line2}.pdbqt --out docking/${line2}_out.pdbqt --log docking/${line2}_out.log &
28     vina --config dock.conf --ligand compounds/${line3}.pdbqt --out docking/${line3}_out.pdbqt --log docking/${line3}_out.log &
29     vina --config dock.conf --ligand compounds/${line4}.pdbqt --out docking/${line4}_out.pdbqt --log docking/${line4}_out.log &
30     vina --config dock.conf --ligand compounds/${line5}.pdbqt --out docking/${line5}_out.pdbqt --log docking/${line5}_out.log &
31     vina --config dock.conf --ligand compounds/${line6}.pdbqt --out docking/${line6}_out.pdbqt --log docking/${line6}_out.log &
32     vina --config dock.conf --ligand compounds/${line7}.pdbqt --out docking/${line7}_out.pdbqt --log docking/${line7}_out.log &
33     vina --config dock.conf --ligand compounds/${line8}.pdbqt --out docking/${line8}_out.pdbqt --log docking/${line8}_out.log &
34     vina --config dock.conf --ligand compounds/${line9}.pdbqt --out docking/${line9}_out.pdbqt --log docking/${line9}_out.log &
35     vina --config dock.conf --ligand compounds/${line10}.pdbqt --out docking/${line10}_out.pdbqt --log docking/${line10}_out.log &
36     wait
37 done
38
39 date
```

- First line is always the shebang
- Using software installed on OSCER
- Makes docking folder
- Open the file job\_id.index, read 10 lines, run vina for each line in the background, and then wait for all jobs to finish before running the next 10 lines.

- SLURM batch commands for submitting jobs
- Partition - group of nodes based on hardware (debug is for short 30 min jobs)
- Exclusive - request entire node to yourself
- ntask - # of tasks expected
- Output/error - gives output/error file with job ID as name
- Time - requested time

## **gnu\_parallel\_2cores (~11 min)**

---

- From the no\_external\_lib folder, we can go to this folder by:
  - Cd .../gnu\_parallel\_2cores
- Prepare the compound list by:
  - Bash setup.sh
- Modify the dock.conf for the search volume, CPU, and location of the protein file
  - Could be copied from the previous folder by: cp ..../no\_external\_lib/dock.conf .
- Submit the SLURM script:
  - Sbatch runparallel.slurm

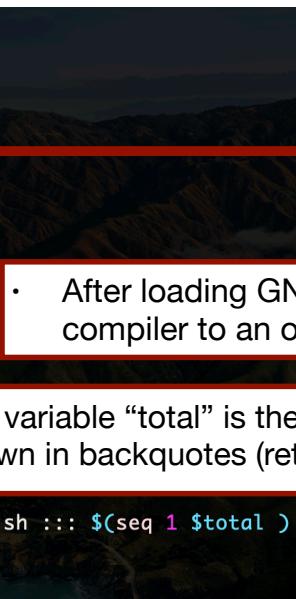
## gnu\_parallel\_2cores (runparallel.slurm - the submitting script)

- First line is always the shebang

```
1 #!/bin/bash
2 #SBATCH --partition=debug
3 #SBATCH --exclusive
4 #SBATCH --ntasks=20
5 #SBATCH --output=%j.out
6 #SBATCH --error=%j.err
7 #SBATCH --time=00:30:00
8 #SBATCH --job-name=vs
9
10 date
11
12 module load parallel/20190222-GCCcore-7.3.0
13 module load GCCcore/4.9.3
14 module load AutoDockVina/1.1.2
15
16 total=`ls compounds/*.pdbqt | wc -l`
17
18 mkdir -p docking
19
20 parallel --progress --jobs 10 bash rundocking.sh ::: $(seq 1 $total )
21
22 date
23
```

- Makes docking folder

- Parallel is the program that will manage our serial jobs.
  - progress : job progress information printed in the SLURM error file
  - jobs # : run up to # jobs in parallel
  - Bash rundocking.sh : execute the rundocking.sh script
  - ::: \$seq( 1 \$total) argument for parallel is a sequence of numbers from 1 to the variable total



- After loading GNU Parallel, change the compiler to an older version for Vina

- The variable "total" is the output of commands shown in backquotes (returns single number)

- SLURM batch commands for submitting jobs
  - Partition - group of nodes based on hardware (debug is for short 30 min jobs)
  - Exclusive - request entire node to yourself
  - ntask - # of tasks expected
  - Output/error - gives output/error file with job ID as name
  - Time - requested time

## gnu\_parallel\_2cores (rundocking.sh - the docking script)

- First line is always the shebang

```
1 #!/bin/bash
2
3 compound=`awk "NR==$1" job_id.index`
4
5 vina --config dock.conf --ligand compounds/${compound}.pdbqt --out docking/${compound}_out.pdbqt --log docking/${compound}_out.log
6
```

- The variable “compound” is the output of the commands in backquotes (returns the compound name based on line number)
- AWK is a function for text/data processing, NR is built in AWK variable to return line (NR = number of records)
- \$1 is the first argument given to the bash script
- i.e. In the runparallel.slurm script, the first job ran:
  - bash rundocking.sh 1 ( bash <script> <argument 1> .. )