AutoDockFR – manual

A. Rigid receptor docking.

1. Prepare the receptor and ligand as described in *AutoDock* manual.
2. Generate maps using *Autogrid* providing ligand atom types as described in *AutoDock* manual.
3. Post-process the maps using *fixmap.py* script in ADFR.
    a. Creates maps with a potential gradient from the center to the surface of the protein.
    b. Generates translational points.
    c. Example setting file:
    ```
    Ligand = './1ykr/1YKR_lig.pdbqt'
    Receptor = './1ykr/4EK3_rec.pdbqt'
    mapECutOff = -0.30
    gridMaps = ['./1ykr/4EK3_rec.Cl.map', './1ykr/4EK3_rec.e.map', './1ykr/4EK3_rec.HD.map', './1ykr/4EK3_rec.OA.map',
    './1ykr/4EK3_rec.d.map', './1ykr/4EK3_rec.A.map', './1ykr/4EK3_rec.C.map', './1ykr/4EK3_rec.NA.map',
    './1ykr/4EK3_rec.N.map', './1ykr/4EK3_rec.S.map']
    fixMaps = True
    transPointsFile = '4EK3_rec_anchorGenes-0.30.py'
    coarseStep = 3
    ```
    d. Run command: ***pythonsh AutoDockFR/bin/fixMaps.py –p settings***
4. Prepare the setting file for the docking. Sample setting file can be found below.

```
jobID= '_rigid_1'
### Ligand File ###
Ligand= '1YKR_random.pdbqt'
### Receptor File ###
Receptor = '4EK3_rec.pdbqt'
ReceptorCONECT = False
gridMaps = ['./1ykr/4EK3_rec_fixedWith-0.30.A.map', './1ykr/4EK3_rec_fixedWith-0.30.S.map', './1ykr/4EK3_rec_fixedWith-
0.30.HD.map', './1ykr/4EK3_rec_fixedWith-0.30.d.map', './1ykr/4EK3_rec_fixedWith-0.30.Cl.map', './1ykr/4EK3_rec_fixedWith-
0.30.C.map', './1ykr/4EK3_rec_fixedWith-0.30.NA.map', './1ykr/4EK3_rec_fixedWith-0.30.e.map', './1ykr/4EK3_rec_fixedWith-
0.30.N.map', './1ykr/4EK3_rec_fixedWith-0.30.OA.map']
mapECutOff = -0.30
### Docking Search Parameters ###
search = 'GA2_1'
# Number of docking runs
constraintMaxTry = 0
GAminimize = {'nbSteps':1, 'noImproveStop':1, 'max_steps':100, 'MAX_FAIL':3, 'MIN_VAR':0.01}
GAHighMinimize = {'nbSteps':10, 'noImproveStop':3, 'max_steps':300, 'MAX_FAIL':4, 'MIN_VAR':0.01, 'search_rate':0.05}
#repeat = 1
# RMSD structure
rmsdRef=['./1ykr/1YKR_lig.pdbqt']
rmsdSym = 'HungarianMatching'
transPointsFile = '4EK3_rec_anchorGenes-0.30.py'
### GA settings ###
rand_seed = 2014
GA_optFEB = 0
GA_gens = 1
GA_pop_size = 10
GA_replace = 0.8
GA_injectRandomInd = 0.
GA_crossover = 0.8
GA_mutation = 0.05
GA_deviation = 0.0001
GA_max_eval = 25000000
## GA local Search parameters
#enable the local search? True, False
GA_enableLocalSearch = True
#local search probability
GA_localsearchfreq = 0.00
GA_localsearchrate = 0.3
fixedLigandRotationAndTranslation = False
fixedLigandConformation = False
AnnealSteps = 0
### ScoringFunction Parameters ###
```

```
# C++ scorer or scoringFunction
scoringFunction = "AutoDock4.2"
# Create scorers
RR_L = True
FR_L = False
L_L = True
RR_RR = False
RR_FR =False
FR_FR =False
# Scorers that contribue to GA fitness function
RR_L_Fitness = True
FR_L_Fitness = False
L_L_Fitness = True
RR_RR_Fitness = False
RR_FR_Fitness = False
FR_FR_Fitness = False
```

5. Run AutoDockFR: ***pythonsh AutoDockFR/bin/runadfr.py –p settings***

B. Flexible receptor docking

1. Prepare the receptor and ligand as described in *AutoDock* manual.
2. Generate maps using *Autogrid*, providing ligand and moving receptor atom types.
3. Create rigid receptor by removing the moving side-chain atoms along with their C_α and C_β. Use *flexResprep.py* script provided in the scripts folder. Required input : receptor pdbqt file and flexible residues as selection string. Command: *pythonsh flexResprep.py -rec <rec.pdbqt> –flexRes <‘rec_name:Chain:RESXXX’> –outDir <path>’*. Output files will be rigid receptor pdbqt without C_α and C_β of the flexible residues (recNoSC.pdbqt), and flexible moving atoms of the flexible residues as a pdbqt file (recFlexres.pdbqt).
4. Post-process the maps using *fixmap.py* script in ADFR with 1) full receptor and 2) with rigid receptor.
    a. Generated translational points from the carbon map from (1) is used for flexible docking. Setting file as in A. 3c is used.
    b. The processed maps from (2) should be used for flexible docking.
    c. Example setting file for 4b.

    ```
    Ligand = './1ykr/1YKR_lig.pdbqt'
    Receptor = './1ykr/4EK3_recNoSC.pdbqt'
    mapECutOff = -0.30
    gridMaps = ['./1ykr/4EK3_recNoSC.Cl.map', './1ykr/4EK3_recNoSC.e.map', './1ykr/4EK3_recNoSC.HD.map',
    './1ykr/4EK3_recNoSC.OA.map', './1ykr/4EK3_recNoSC.d.map', './1ykr/4EK3_recNoSC.A.map',
    './1ykr/4EK3_recNoSC.C.map', './1ykr/4EK3_recNoSC.NA.map', './1ykr/4EK3_recNoSC.N.map',
    './1ykr/4EK3_recNoSC.S.map']
    fixMaps = True
    transPointsFile = '4EK3_recNoSC_anchorGenes-0.30.py'
    coarseStep = 3
    ```

    d. Run command: ***pythonsh AutoDockFR/bin/fixMaps.py –p settings***
5. Prepare the setting file for the docking. Sample setting file can be found below.

```
jobID= '_flex_1'
### Ligand File ###
Ligand= '1YKR_random.pdbqt'
### Receptor File ###
Receptor = '4EK3_rec.pdbqt'
ReceptorCONECT = False
gridMaps = ['./1ykr/4EK3_recNoSC_fixedWith-0.30.OA.map', './1ykr/4EK3_recNoSC_fixedWith-0.30.S.map',
'./1ykr/4EK3_recNoSC_fixedWith-0.30.NA.map', './1ykr/4EK3_recNoSC_fixedWith-0.30.N.map', './1ykr/4EK3_recNoSC_fixedWith-
0.30.C.map', './1ykr/4EK3_recNoSC_fixedWith-0.30.e.map', './1ykr/4EK3_recNoSC_fixedWith-0.30.d.map',
'./1ykr/4EK3_recNoSC_fixedWith-0.30.Cl.map', './1ykr/4EK3_recNoSC_fixedWith-0.30.HD.map', './1ykr/4EK3_recNoSC_fixedWith-
0.30.A.map']
mapECutOff = -0.30
### Docking Search Parameters ###
search = 'GA2_1'
# Number of docking runs
```

```
constraintMaxTry = 0
GAminimize = {'nbSteps':1, 'noImproveStop':1, 'max_steps':100, 'MAX_FAIL':3, 'MIN_VAR':0.01}
GAHighMinimize = {'nbSteps':10, 'noImproveStop':3, 'max_steps':300, 'MAX_FAIL':4, 'MIN_VAR':0.01, 'search_rate':0.05}
#repeat = 1
# RMSD structure
rmsdRef=['./1ykr/1YKR_lig.pdbqt']
rmsdSym = 'HungarianMatching'
transPointsFile = '4EK3_rec_anchorGenes-0.30.py'
movingSC = '4EK3_rec:A:LYS33,LYS89'
rmsdRecRef=['./1ykr/4EK3_recFlexres.pdbqt']
### GA settings ###
rand_seed = 2014
GA_optFEB = 0
GA_gens = 1
GA_pop_size = 10
GA_replace = 0.8
GA_injectRandomInd = 0.
GA_crossover = 0.8
GA_mutation = 0.05
GA_deviation = 0.0001
#large number (GA_pop_size * GA_gens = # of possible evals.  This doesnt count the local search evals either...)
GA_max_eval = 25000000
## GA local Search parameters
#enable the local search? True, False
GA_enableLocalSearch = True
#local search probability
GA_localsearchfreq = 0.00
GA_localsearchrate = 0.3
fixedLigandRotationAndTranslation = False
fixedLigandConformation = False
AnnealSteps = 0
### ScoringFunction Parameters ###
# C++ scorer or scoringFunction
scoringFunction = "AutoDock4.2"
# Create scorers
RR_L = True
FR_L = True
L_L = True
RR_RR = False
RR_FR =True
FR_FR =True
# Scorers that contribue to GA fitness function
RR_L_Fitness = True
FR_L_Fitness = True
L_L_Fitness = True
RR_RR_Fitness = False
RR_FR_Fitness = True
FR_FR_Fitness = True
```

6. Run AutoDockFR: ***pythonsh AutoDockFR/bin/runadfr.py –p settings***