

TINKER simulations

Molecular Dynamics Simulations

An EMBnet introductory course

10-11-2008

Jose R. Valverde

EMBnet/CNB, CSIC

<http://www.es.embnet.org>

jrvalverde@cnb.csic.es

- TINKER introduction
- Protein simulations *in vacuo*
- Protein simulations in implicit solvent model
- Solvent simulations with periodic boundary conditions
- Protein simulations in explicit, all-atoms solvent model

TINKER introduction

- Developed at Jay Ponder's lab
 - <http://dasher.wustl.edu/tinker/>
- Professional quality
- Main goal is development of algorithms
- Less extensive than other programs (e. g. CHARMM)
- Wider support for force fields
 - AMBERxx, AMOEBAxx, CHARMMxx, Dang, Dudek, Hoch, MMxx, OPLSxx, SMOOTHxx...
- Focus on ease of use
- Command line interface
- Free, with source code
- Integrated with GAMESS-US QM software (SIMOMM)

- Coordinates (XYZ)
 - Convert from/to PDB (**pdboxyz**, **xyzpdb**)
- Parameter file
 - .key file
 - **tinker.key** (always valid)
 - **filename.key** (*filename* is same as in *filename.xyz*)
 - keyword value
 - **parameters /opt/structure/tinker/params/amber.prm**
- Other files
 - Parameter files (potential energy functions/force fields)
 - Restart, output, trajectories
- Interactive command line
 - Easy: just answer
 - Answers may be given in the command line

Worth noting

- Filename extension is not entered (tinker assumes it)
- Tinker asks for options not given in the command line
- Tinker outputs progress report to console
 - Worth saving for inspection/analysis
 - `command model | tee model-command.log`
 - `command model > model-command.log &`
- *filename.key* takes precedence over *tinker.key*
 - Use *tinker.key* for common options
 - User *filename.key* for model-specific options
- Existing output files are not overwritten
 - A sequential suffix is added (e. g. `model.xyz_2`)

Protein model *in vacuo*

- Create a subdirectory and move to it
 - Copy PDB coordinate file to your directory
 - e. g. **2ech.pdb**
 - Look into tinker installation directory, folder 'params'
 - e. g. **ls /opt/structure/tinker/params/**
 - Select a force field and put it into **tinker.key**
 - e. g.
parameters /opt/structure/tinker/params/charmm27.prm
 - Convert PDB coordinates to XYZ coordinates
 - e. g. **pdbxyz 2ech**
 - Correct any discrepancies in atom names
 - PDB files tend to use atom names corresponding to the force field used during refinement
 - Not all force fields use the same atom names
- Have an initial look with **analyze**

Initial structure optimization

- Tinker offers various methods
 - minimize
 - optimize
 - newton
 - PSS
 - sniffer
- Minimization progresses until a target RMSD is reached
 - Use target RMSD of 1.0
 - Use chosen program interactively or in batch mode
 - `minimize 2ech 1.0 | tee 2ech-minimize.log`
 - `newton 2ech A A 1.0 > 2ech-newton.log &`
- Speed up using cutoffs in tinker.key
 - cutoff 15.0

- Tune computation in `tinker.key`
 - cutoff 15.0
 - lights
- Reduce output clutter in `tinker.key`
 - archive
- Run ANNEAL to heat progressively the system to desired temperature
 - Heat from 1 to 298 K
 - Do not perform previous equilibration (0 steps)
 - Apply heat linearly
 - Use a 1.0 fs timestep and save data every 0.1 ps
 - Keep atomic weights
 - Run in batch mode
 - `anneal 2ech 1 298 0 2000 L 1.0 0.1 0.0 > 2ech-anneal.log &`

- Keep the system at target temperature until equilibrium is reached
- Run DYNAMIC to simulate 3ps at 298K and monitor energies
 - Continue using same parameters
 - Run in batch mode
 - `dynamic 2ech 3000 1.0 0.1 298 > 2ech-equil.out &`
 - After completion check system evolution
 - `grep 'Temperature' 2ech-equil.out > temperature.data`
 - `grep 'Total Energy' 2ech-equil.out > totalenergy.data`
 - ...
 - Import into a graphical program and plot fluctuations
 - Visualize the trajectory
 - e. g. with gopenmol

- Gather thermodynamical statistics and trajectory
- Start from equilibrated structure
- Run DYNAMIC to simulate 5 ps at 298 K
 - Timestep: 1fs
 - Number of steps: 5000
 - Take snapshots every 100 steps (100 fs = 0.1 ps)
 - Run in batch mode
 - `dynamic 2ech 5000 1.0 0.1 298 > 2ech-prod.log &`
- Monitor progress
 - Visualize trajectory
 - Watch out for strong energy fluctuations

- Visualize trajectory
- Extract relevant data
 - `grep 'Total Energy' 2ech-prod.log > te.out`
 - `cat te.out | sed -e 's/ Total Energy//g' > te.1`
 - `cat te.1 | sed -e 's/K.*//g' > te.2`
 - `cat te.2 | sed -e 's/ //g' > te.dat`
- Plot data
 - Import into plotting program
- Perform distance geometry calculations
 - DISTGEOM
- Compute time dependent correlations
 - CORRELATE

Protein in implicit solvent model

Define implicit solvent model

- Select an implicit solvation model
 - ASP / SASA / ONION / STILL / HCT / ACE / GBSA
 - Add it to **tinker.key**
 - **solvate still**
 - State solvent dielectric constant
 - **dielectric 80.0**
 - State polarization effects
 - **polarization direct**
- Perform the whole simulation process using the new parameters (tip: use a separate directory)
 - Start from initial PDB
 - Minimize
 - Equilibrate
 - Production run
 - Analyze



Easy, wasn't it?

Water in Periodic Boundary Conditions

- Use a fresh directory
- Start with the coordinates for one molecule of water
 - water.xyz
- Choose a model for water (e. g. TIP3P)
 - CHARMM contains a TIP3P model
 - You can build your own
 - Check atom name discrepancies
 - e. g. for CHARMM you get O \rightarrow OT (101), H \rightarrow HT (88)
- Generate a water box using XYZEDIT
 - **xyzedit water**
 - Place 27 molecules in a cubic box of 9.3125 Å side

- Define periodic boundary conditions in `tinker.key` (or in `water.key`):
 - Box size
 - `a-axis 9.3125`
 - Cutoff distance (less than $\frac{1}{2}$ the box size)
 - `cutoff 4.0`
- Fine tune computation (TIP3P is a rigid model)
 - `integrate RIGIDBODY`
 - `group-molecule`
 - `neutral-groups`
- Heat the system during 6 ps at NVT.
 - `dynamic waterbox 6000 1.0 0.1 2 298`
 - `anneal waterbox 1 298 0 6000 L 1.0 0.1 0.0`

- After we have heated the system, we let it equilibrate for another 6ps at 298K (NVT)
 - `dynamic waterbox 6000 1.0 0.1 2 298`
- Analyze system evolution for equilibration
 - `grep 'Temperature' water-equil.out > temperature.data`
 - `grep 'Total Energy' water-equil.out > totalenergy.data`
 - ...
 - Import into a graphical program and plot fluctuations
- Visualize the trajectory
 - e. g. with gopenmol

- Gather thermodynamical statistics and trajectory
- Start from equilibrated structure
- Run DYNAMIC to simulate 5 ps in the NVE ensemble
 - Timestep: 1fs
 - Number of steps: 5000
 - Take snapshots every 100 steps (100 fs = 0.1 ps)
 - Run in batch mode
 - `dynamic waterbox 5000 1.0 0.1 1 > production.log &`
- Monitor progress
 - Visualize trajectory
 - Watch out for strong energy fluctuations
- Analyze results.

Protein in explicit solvent model

- Explicit solvent models are very expensive
 - Approximations and shortcuts have dramatic effects
 - Boundary Conditions are required to keep the solvent confined
 - Periodic Boundary Conditions (spherical)
 - Stochastic Boundary Conditions (Brownian motion)
 - Cutoffs and cell sizes reduce computation area
 - Reaction region
 - Buffer (Langevin motion) region
 - Reservoir (fixed) region

- Preparation
 - Generate biomolecule coordinates input file
 - Compute energy
 - Minimize Energy
- Generate solvation sphere
 - Generate solvent cube
 - Soak biomolecule in solvent
 - Minimize water around biomolecule
 - Equilibrate water around biomolecule
- Heat-Equilibration-Production of entire system
- Analyze results.

- Same as we do for *in vacuo* simulations:
 - Grab PDB file
 - Convert to XYZ
 - Analyze structure
 - Minimize structure energy
- You can start from saved minimized coordinates from a previous *in vacuo* run.

- Create a box of water
 - Get hold of a water molecule
 - Select force field (e. g. CHARMM27)
 - Fix atom name and definition discrepancies
 - e.g. O : OHT, type 101, H : HT, type 88
 - Run XYZEDIT
 - Number of molecules: 1600
 - Box size: 36.342
 - Translate box to center of mass using XYZEDIT
- Solvate biomolecule
 - Open with XYZEDIT
 - Translate to center of mass
 - Soak in water
 - No need to remove water outside the sphere
 - TINKER will ignore it once defined.

Equilibration of water around fixed protein

- Open original biomolecule file
 - Note down molecule size (713 atoms)
- Define computation conditions in tinker.key
 - Use spherical boundary conditions
 - The system is centered around origin 0.0.0
 - The system size is ~32 Å
 - sphere 0.0 0.0 0.0 18.0
 - wall
 - Exclude biomolecule by defining active region
 - Range from first (714) to last (5372) solvent atoms
 - active -714 5372
 - Stir the bottle to remove strong Van der Waals interactions
 - rattle
- Minimize energy and equilibrate as usual.

- Once the solvent has adapted to the biomolecule's environment we can run the simulation:
 - Remove restriction on the active region
 - Select a faster integration method
 - integrate verlet
- Heat to 298K
 - dynamic 2ech 1000 1.0 0.1 298
 - anneal 2ech 1 298 0 L 1000 1.0 0.1 0.0
- Equilibrate at 298K
 - dynamic 2ech 1000 1.0 0.1 298
- Production run at 298K
 - dynamic 2ech 5000 1.0 0.1 298
- Analyze