Molecular Biophysics & Biochemistry 447b3 / 747b3

Bioinformatics

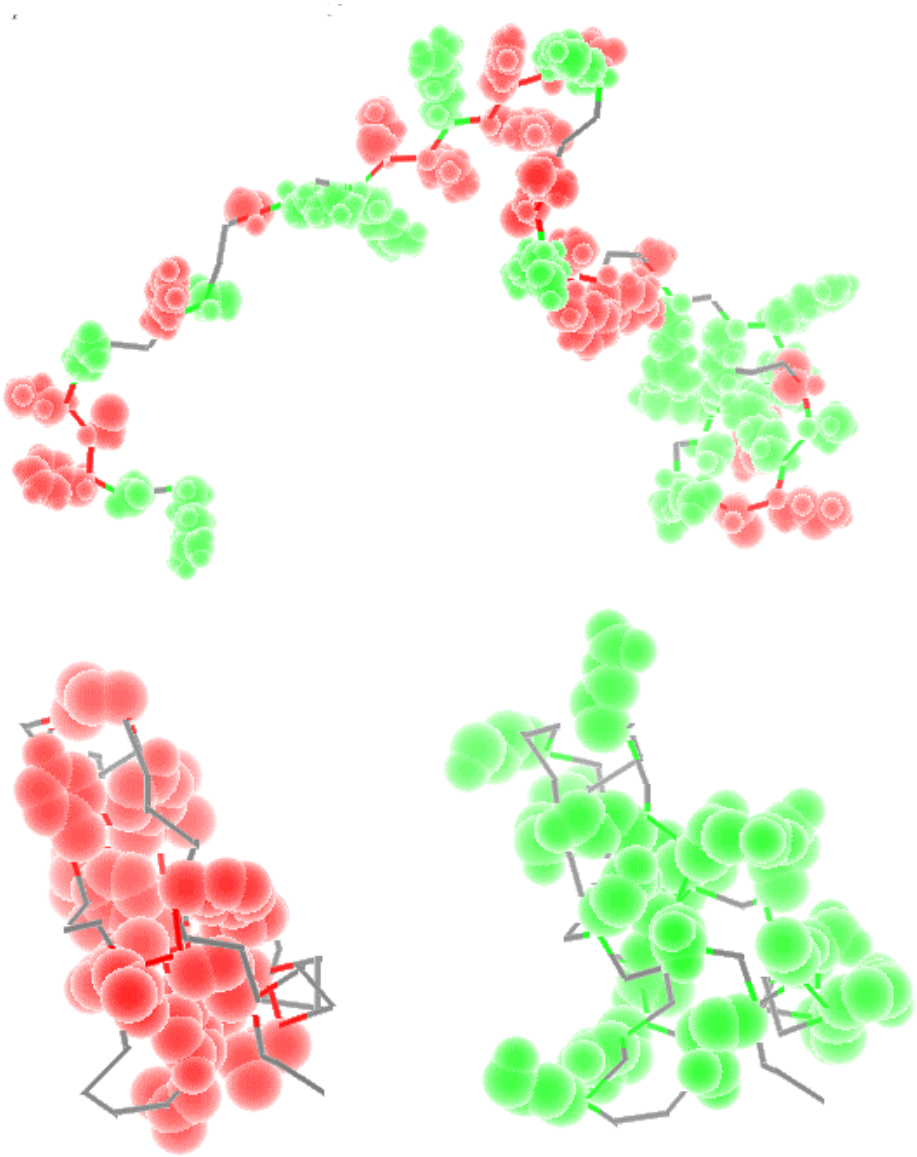
Simulation

Mark Gerstein

Class 13, 2/23/98

Yale University

Goal: Model Proteins and Nucleic Acids as Real Physical Molecules



Overview: Methods for the Generation and Analysis of Macromolecular Simulations

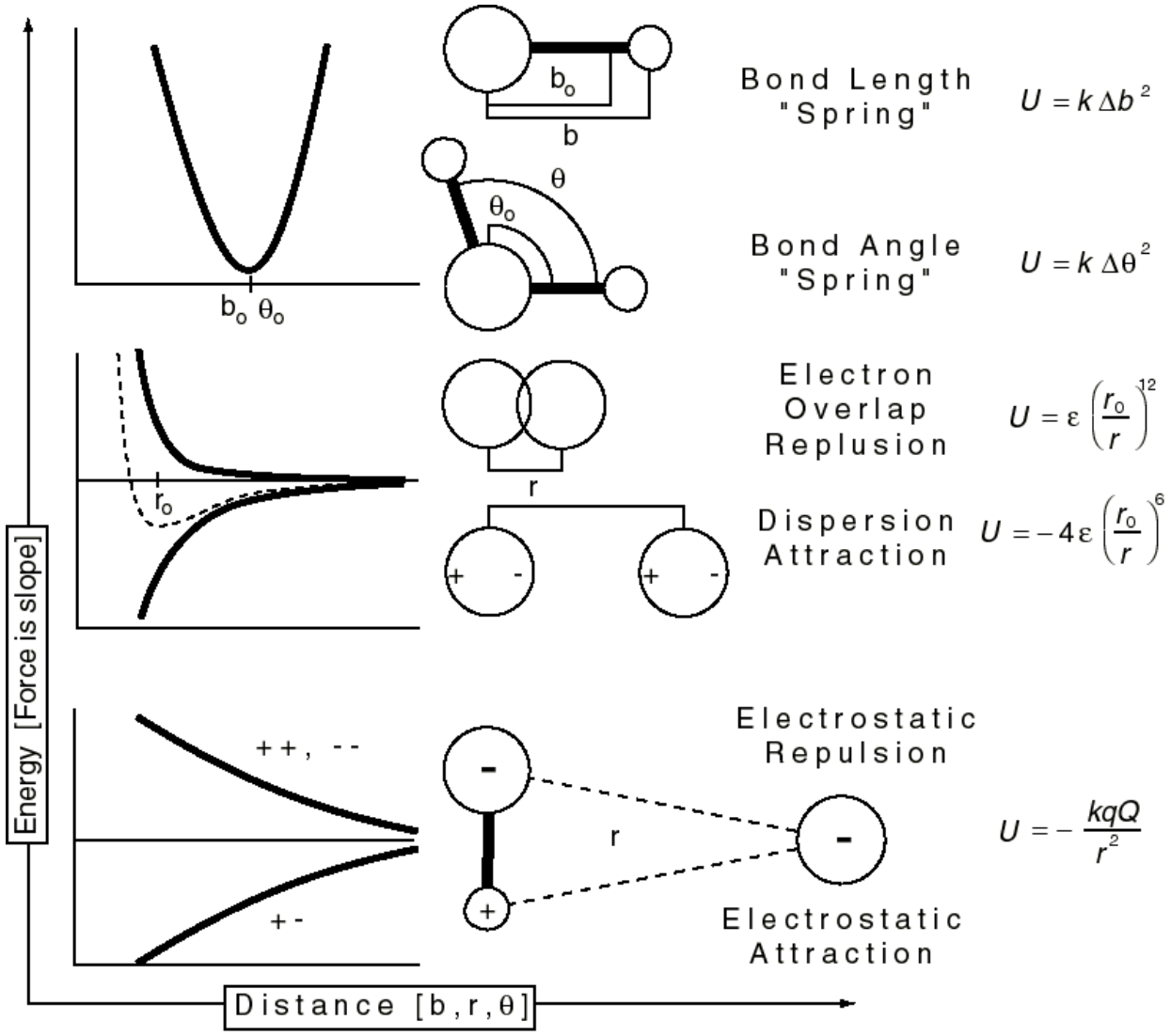
1 Simulation Methods
- ◊ Potential Functions
- ◊ Minimization
- ◊ Molecular Dynamics
- ◊ Monte Carlo
- ◊ Simulated Annealing

2 Types of Analysis
- ◊ liquids: RDFs, Diffusion constants
- ◊ proteins: RMS, Volumes, Surfaces

- Established Techniques (chemisty, biology, physics)
- Focus on simple systems first (liquids). Then explain how extended to proteins.

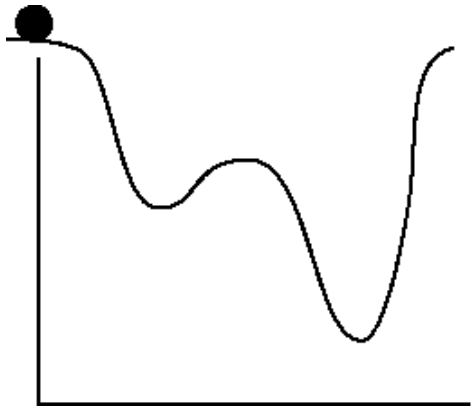
Potential Functions

- Each atom is a point mass (m and **x**)
- Atoms interact through a variety of forces
- Also, for proteins there some special pseudo-forces: torsions and improper torsions, H-bonds.



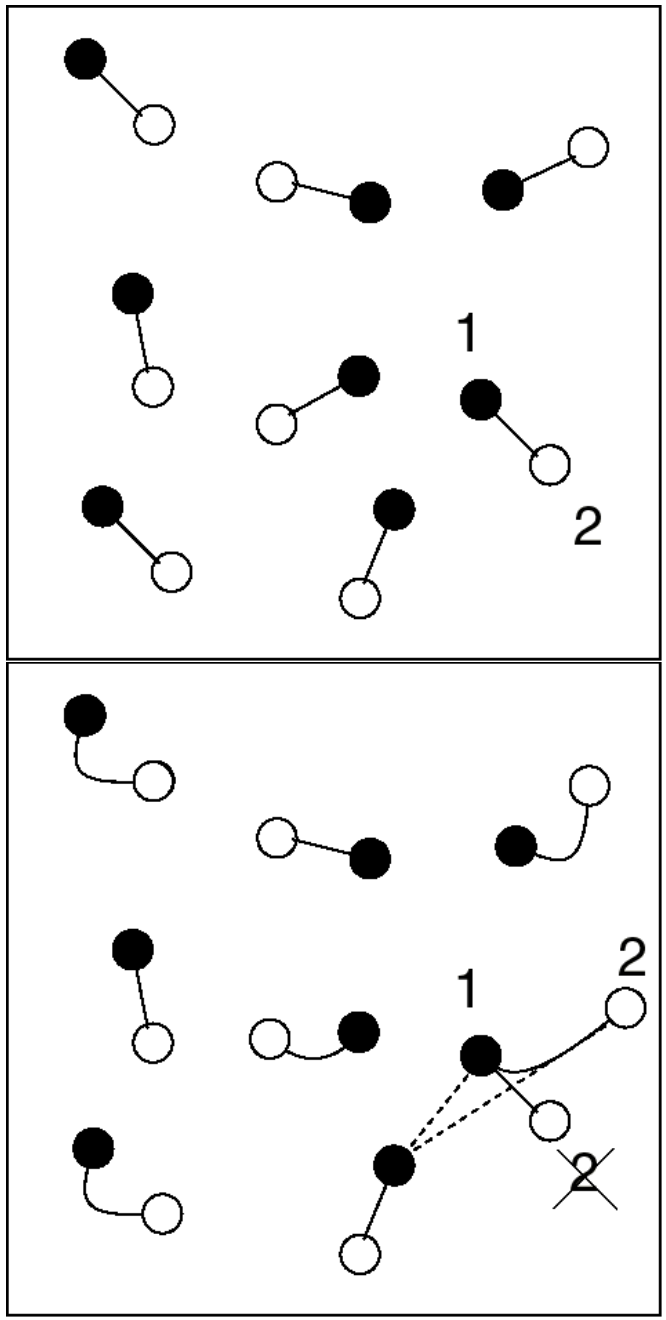
Minimization

- Particles on an "energy landscape." Search for minimum energy configuration
- Steepest descent minimization
  - ◊ Follow gradient of energy straight downhill
  - ◊ i.e. Follow the force:
    step ~ **F** = -∇ U
    so
    x(t) = **x**(t-1) + a **F**/|**F**|
- Other methods
  - ◊ conjugate gradient
    step ~ F(t) **-** b**F**(t-1)
  - ◊ Newton-Raphson:
    using 2nd derivative, find minimum assuming it is parabolic
- Get stuck in local minima



Molecular Dynamics

- Give each atoms a velocity.
  - ◊ If no forces, new position of atom (at t + dt) would be determined only by velocity
    $\mathbf{x}(t+dt) = \mathbf{x}(t) + \mathbf{v}\, dt$
- Forces change the velocity, complicating things immensely
  - ◊ $\mathbf{F} = d\mathbf{p}/dt = m\, d\mathbf{v}/dt$



Molecular Dynamics (cont)

- On computer make very small steps so force is nearly constant and velocity change can be calculated (uniform a)

$$\Delta \mathbf{v} = \frac{\mathbf{F}}{m} \Delta t$$

[Avg. **v** over Δt] = (**v** + Δ**v**/2)

- Trivial to update positions:

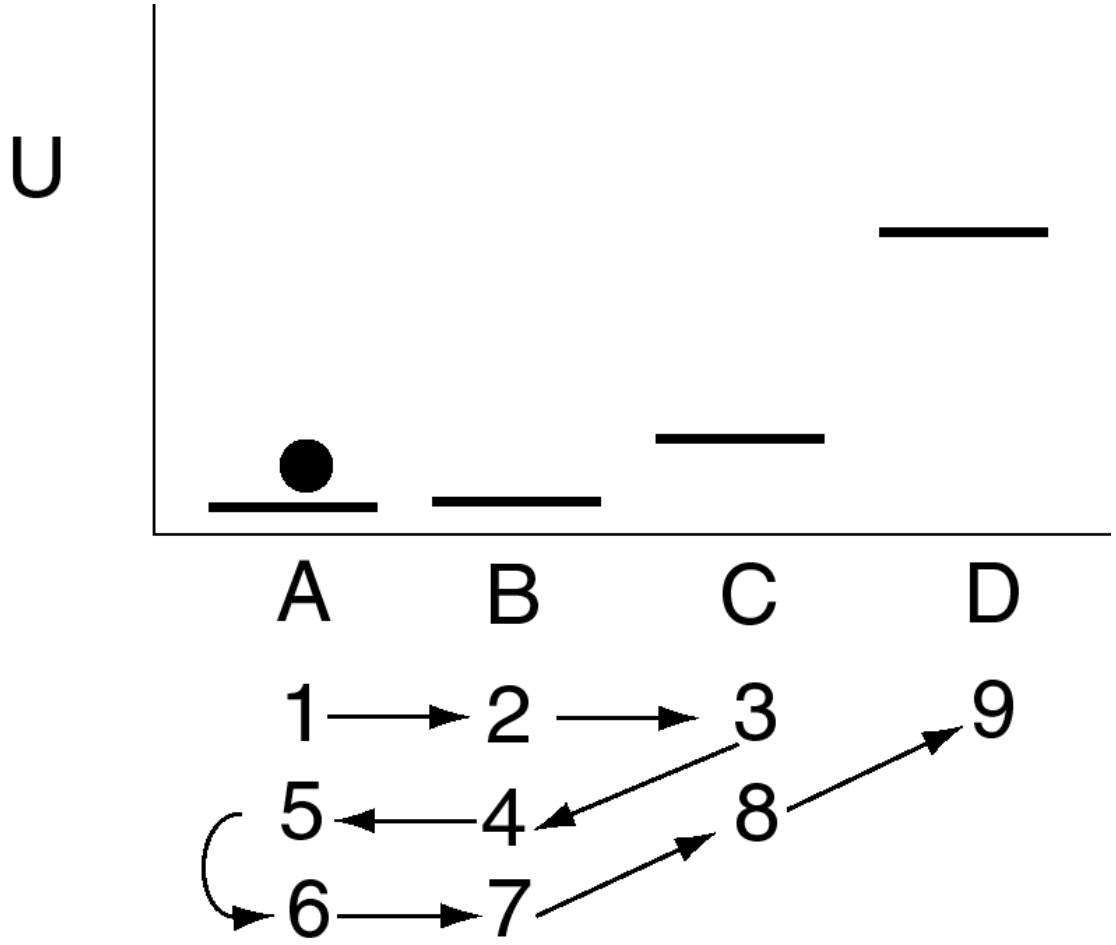
$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + (\mathbf{v} + \frac{\Delta \mathbf{v}}{2}) \Delta t$$

$$= \mathbf{x}(t) + \mathbf{v} \Delta t + \frac{\mathbf{F}}{2m} \Delta t^2$$

- Step must be very small
  - ◊ Δt ~ 1fs (atom moves 1/500 of its diameter)
  - ◊ This is why you need fast computers
- Actual integration schemes slightly more complicated
  - ◊ Verlet (explicit half-step)
  - ◊ Beeman, Gear (higher order terms than acceleration)

Phase Space Walk

- Trajectories of all the particles traverses space of all possible configuration and velocity states (phase space)
- Ergodic Assumption:
  Eventually, trajectory visits every state in phase space
- Boltzmann weighting:
  Throughout, trajectory samples states fairly in terms of system's energy levels
  - ◊ More time in low-U than high-U states
  - ◊ Probability of being in a state ~ exp(-U/kT)
- Consequently, statistics (average properties) over trajectory are thermodynamically correct



Example Phase Space Walk

$$\langle X \rangle = 3X_A + 3X_B + 2X_A + 1X_D$$

$$\langle U \rangle = 6U_{AB} + 2U_A + 1U_D$$

Monte Carlo

- Other ways than MD to sample states fairly and compute correctly weighted averages? Yes, using Monte Carlo calculations.
- Basic Idea: Move through states randomly, accepting or rejecting them so one gets a correct "Boltzmann weighting"

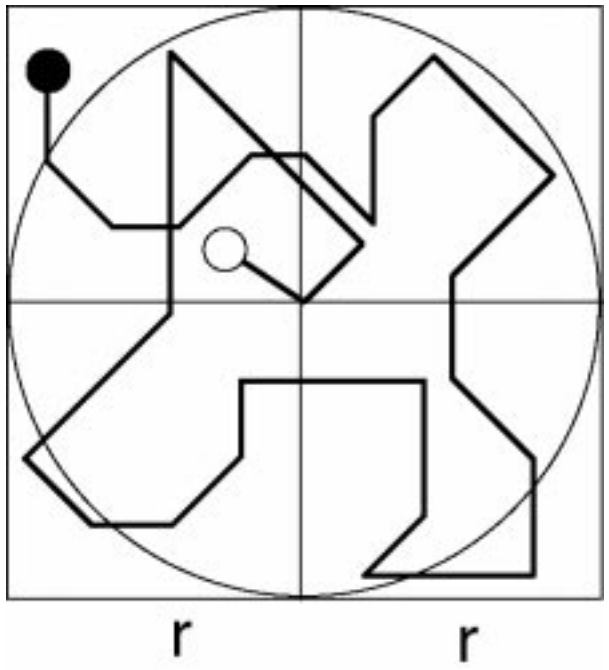
- Formalism:
  - ◊ System described by a probability distribution ρ(n) for it to be in each state n
  - ◊ Random ("Markov") process π operates on the system and changes distribution amongst states to πρ(n)
  - ◊ At equilbrium original distribution and new distribution have to be same as Boltzmann distribution

$$\pi\rho(n) = \rho(n) = \frac{1}{Z}\exp\left(\frac{-U(n)}{kT}\right)$$

Monte Carlo (cont)

- Metropolis Rule (for specifying π)
  1. Make a random move to a particle and calculate the energy change dU
  2. dU < 0 –> accept the move
  3. Otherwise, compute a random number R between 0 and 1:
     R < ~ exp(-U/kT) –>
     accept the move
     otherwise –>
     reject the move

- "Fun" example of MC Integration
  - ◊ Particle in empty box of side 2r (energy of all states same)
  - ◊ π = 6 x [Fraction of times particles is within r of center]

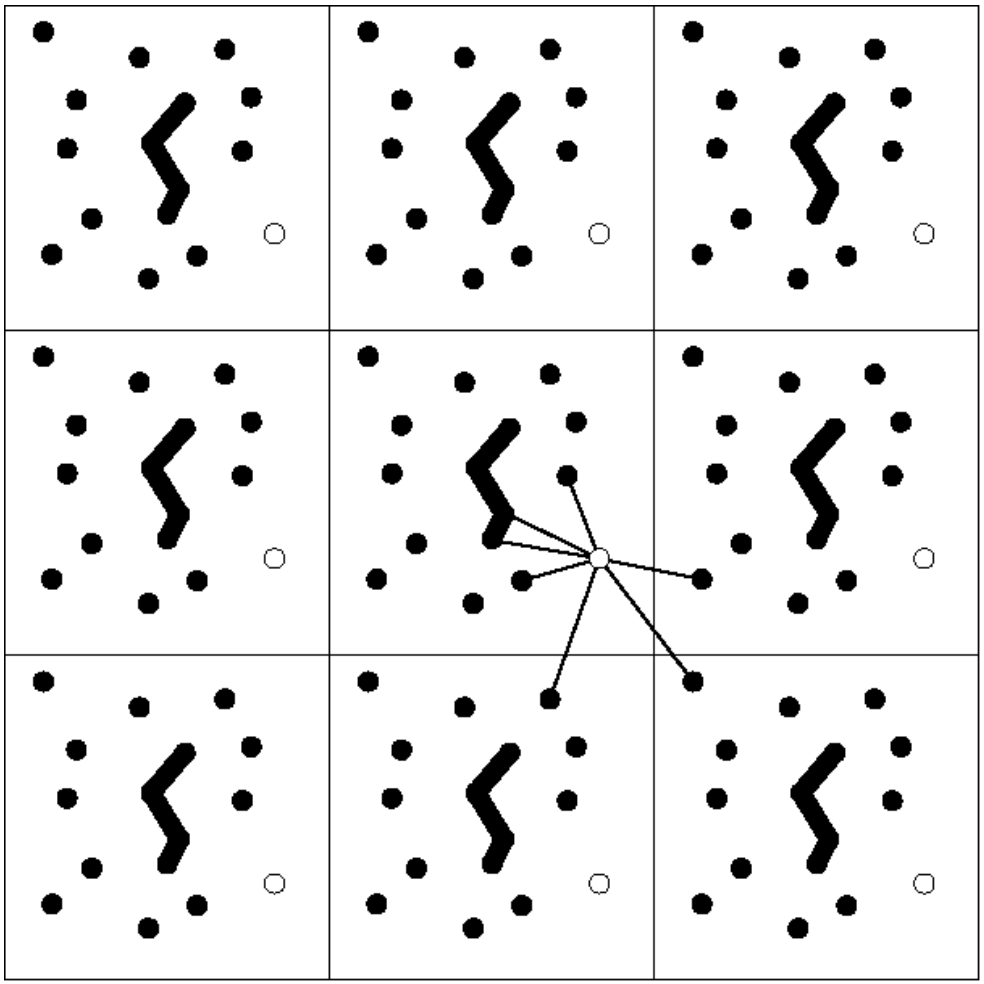


MC vs/+ MD

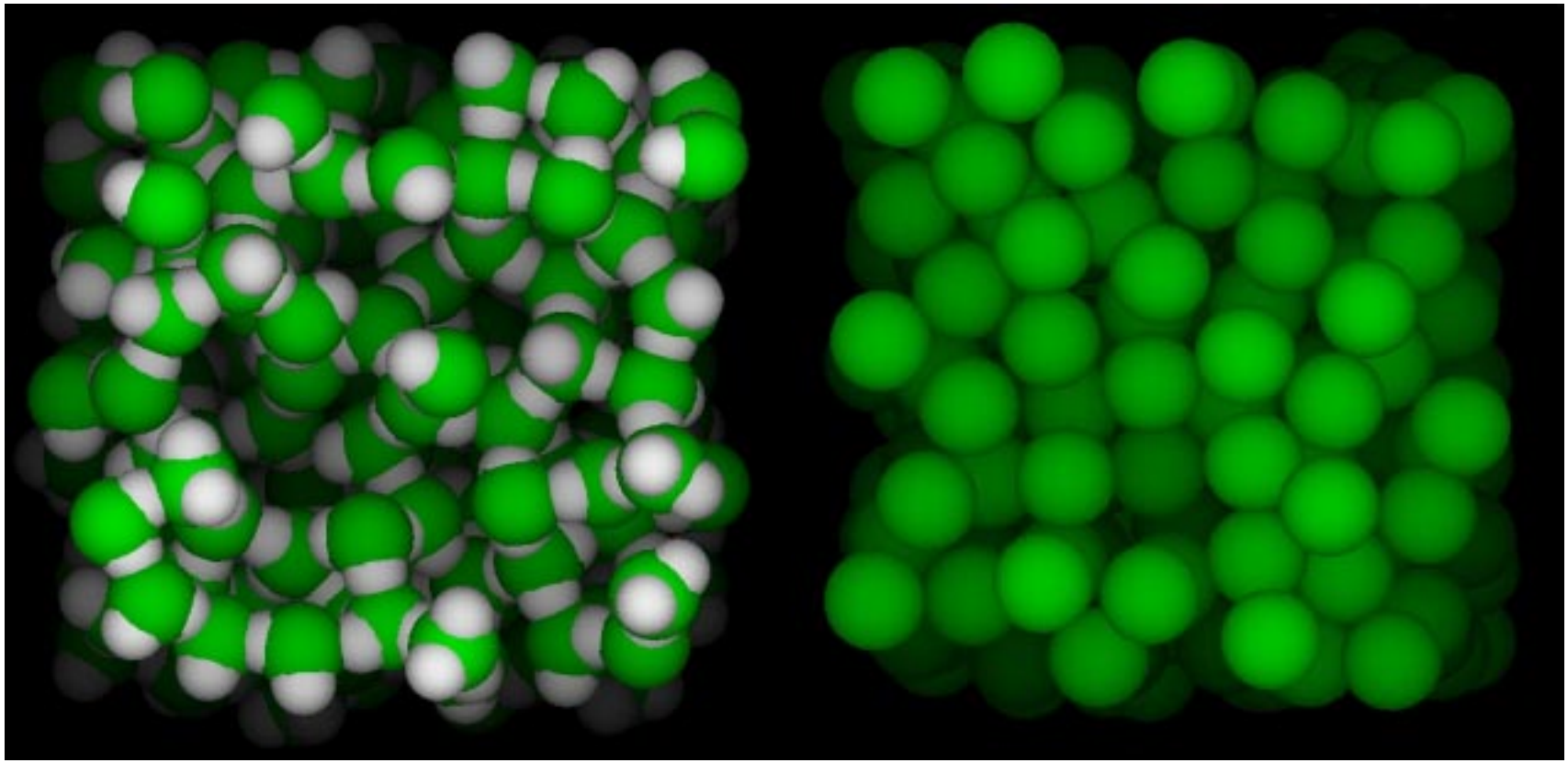
- MD usually used for proteins. Difficult to make moves with complicated chain.
- MC often used for liquids. Can be made into a very efficient sampler.
- Hybrid approaches (Brownian dynamics)
- Simulated Annealing. Heat simulation up to high T then gradually cool and minimize to find global minimum.

Periodic Boundary Conditions

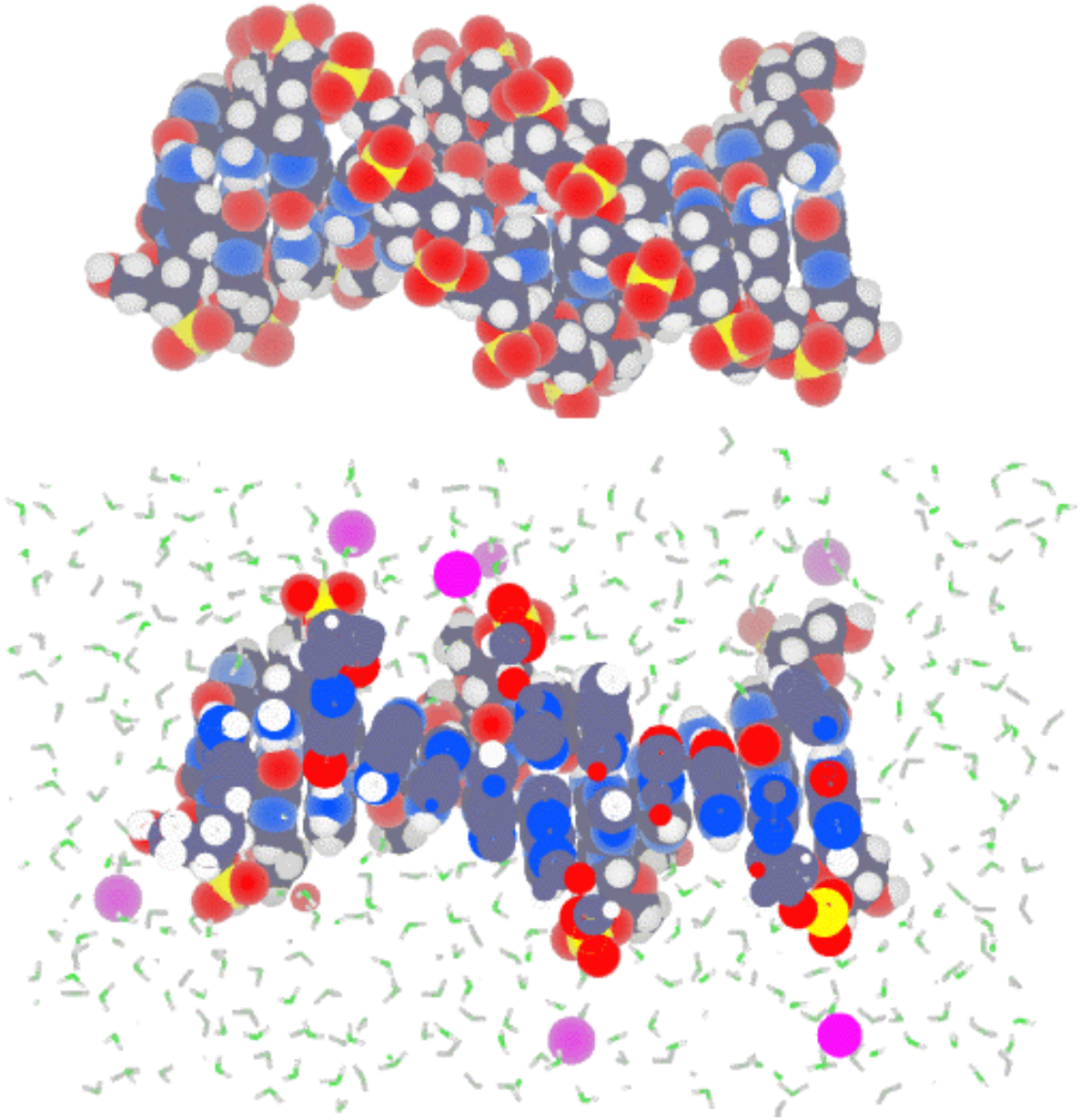
- Make simulation system seem larger than it is



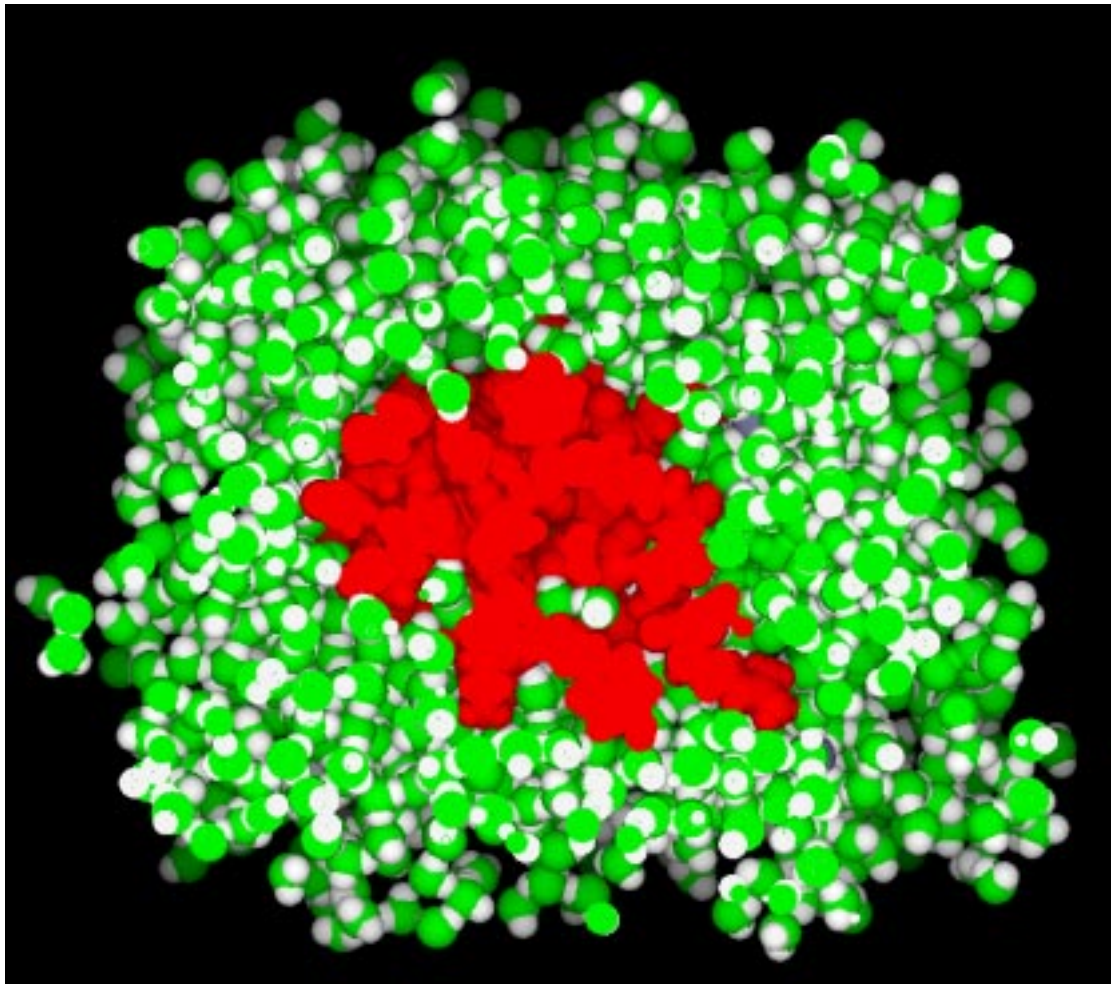
Typical Systems: Water v. Argon



Typical Systems: DNA + Water



Typical Systems: Protein + Water



Average over simulation

- Deceptive Instantaneous Snapshots (almost anything can happen)
- Simple thermodynamic averages
  - ◊ Average potential energy <U>
  - ◊ T ~ < Kinetic Energy > = ½ m < v^2 >
- Some quantities fixed, some fluctuate in different ensembles
  - ◊ NVE protein MD (“microcanonical”)
  - ◊ NVT liquid MC (“canonical”)
  - ◊ NPT more like the real world

Timescales

Motion	length (Å)	time (fs)
bond vibration	0.1	10
water hindered rotation	0.5	1000
surface sidechain rotation	5	10^5
water diffusive motion	4	10^5
buried sidechain libration	0.5	10^5
hinge bending of chain	3	10^6
buried sidechain rotation	5	10^{13}
allosteric transition	3	10^{13}
local denaturation	7	10^{14}

(From McCammon & Harvey, Eisenberg & Kauzmann)

D & RMS

- Diffusion constant
  - ◊ Measures average rate of increase in variance of position of the particles
  - ◊ Suitable for liquids, not really for proteins

$$D = \frac{\langle \Delta r^2 \rangle}{6\Delta t}$$

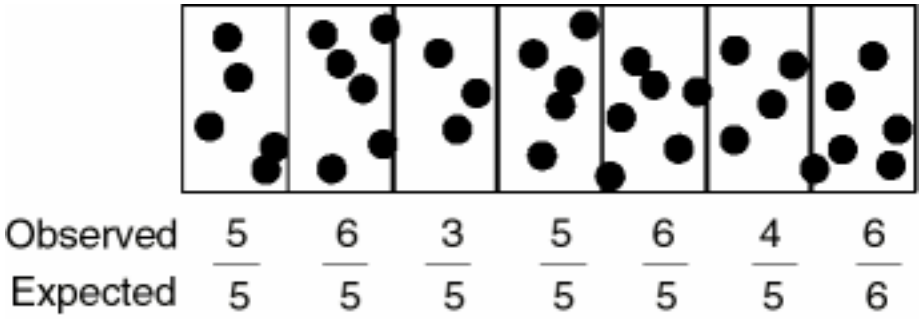
- RMS more suitable to proteins

$$RMS(t) = \sqrt{\frac{\sum_{i=1}^{N} d_i(t)}{N}}$$

$$d_i(t) = \mathbf{R}(\mathbf{x}_i(t) - \mathbf{T}) - \mathbf{x}_i(0)$$

  - o di = Difference in position of protein atom at t from the initial position, after structures have been optimally rotated translated to minimize RMS(t)
  - o Solution of optimal rotation has been solved a number of ways (Kabsch, SVD)

Number Density

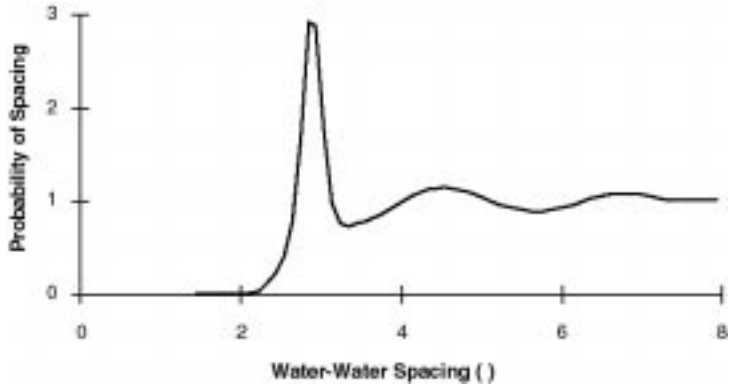


= Number of atoms per unit volume averaged over simulation divided by the number you expect to have in the same volume of an ideal "gas"

Spatially average over all directions gives

1D RDF =

[Avg. Num. Neighbors at r]
[Expected Num. Neighbors at r]



"at r" means contained in a thin shell of thickness dr and radius r

Number Density (cont)

- Advantages: Intuitive, Relates to scattering expts
- D/A: Not applicable to real proteins
  - ◊ 1D RDF not structural
  - ◊ 2D proj. only useful with "toy" systems
- Number densities measure spatial correlations, not packing
  - ◊ Low value does not imply cavities
  - ◊ Complicated by asymmetric molecules
  - ◊ How things pack and fit is property of instantaneous structure - not average