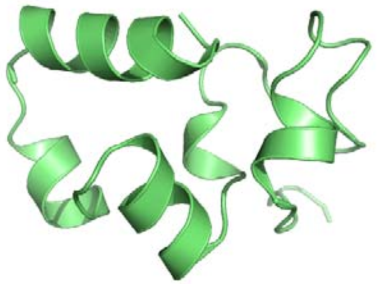
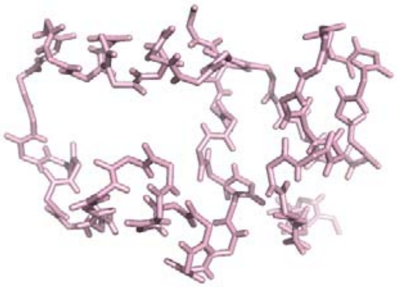
Computational protein design





There are astronomically large number of amino acid sequences that needs to be considered for a protein of moderate size

e.g. if mutating 10 residues, 20^10 = 10 trillion sequences
but each residue has ~ 10 conformational dof
➔ 200^10 ~ Avogadro’s number

Computation can systematically evaluate the quality of different candidate sequences

Computational analysis helps examine the consequence of a perturbation even when experimental validation is difficult or time consuming

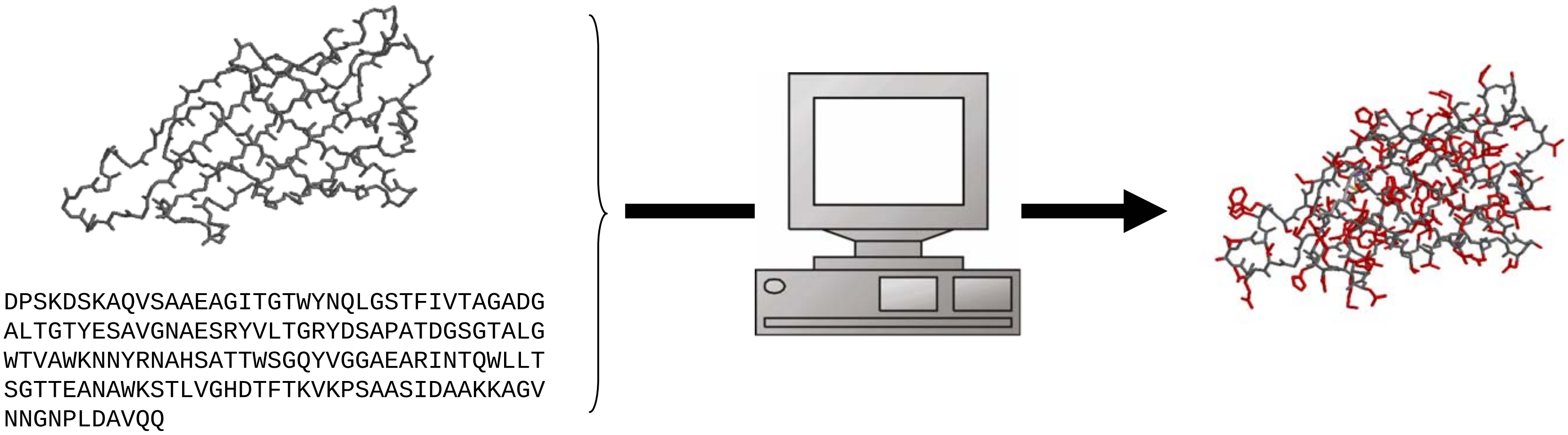
Side chain modeling

Necessary for homology modeling, fold recognition, and folding

Identity and conformation of side chain determine protein stability and its interactions with other molecules

Since the main chain is often assumed constant (or at least assumed known), side chain modeling is a large part of CPD

Computational docking



```
DPSKDSKAQVSAAEAGITGTWYNQLGSTFIVTAGADG
ALTGTYESAVGNAESRYVLTGRYDSAPATDGSGTALG
WTVAWKNNYRNAHSATTWSGQYVGGAEARINTQWLLT
SGTTEANAWKSTLVGHDTFTKVKPSAASIDAAKKAGV
NNGNPLDAVQQ
```

Components of CPD

1. Force field

   compute the energy of a given structure

   different implementations with different functional forms

   special force fields for special occasions, e.g. for membrane proteins

   decide on the resolution—e.g. include hydrogen or not

   solvation effects

2. Amino acid alphabet

   how many different amino acids should be considered

   smaller alphabet may simplify the design and increase the odds of success

3. Amino acid rotamer

   how are amino acids represented computationally

   amino acid has a backbone and a side chain

   side chain conformation should be restricted to a finite number of possibilities

4. Search algorithm

   stochastic or deterministic

   efficiency, convergence

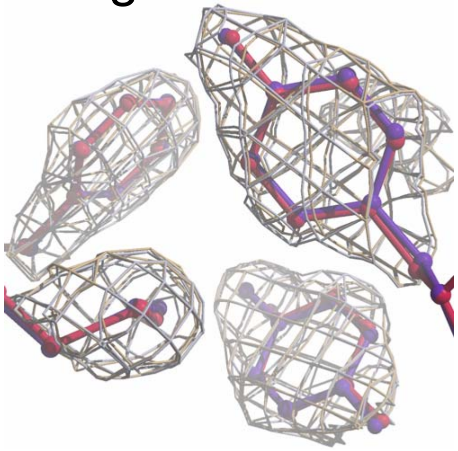
Force fields

The term “force field” is used synonymously with energy function

Relationship between geometric properties and energy (but enthalpy only)

Applications

- compute X-ray and NMR structures consistent with experimental data, e.g. electron density and distance constraints, while minimizing energy
- quantitate the stability of a residue conformation—useful when planning a mutagenesis study
- estimate/evaluate the energy of interaction—i.e. binding affinity—critical during computational drug design
- molecular modeling—either homology/side chain modeling or molecular dynamics and Monte Carlo simulations



Derivation of force fields

Ab initio quantum mechanical calculations

- “first principle” calculation
- solution to the Schroedinger equation
- pros: self-contained, model-independent (other than how one solves the equation), requires few assumptions regarding the functional form
- cons: extremely time consuming, accuracy not as high as desired, severe limitation on the size of the system that can be studied together (~100 atoms), calculation done in gas phase not in condensed matter phase

$$\widehat{H}\psi = E\psi$$

Chemistry Nobel 1998

Empirical force fields

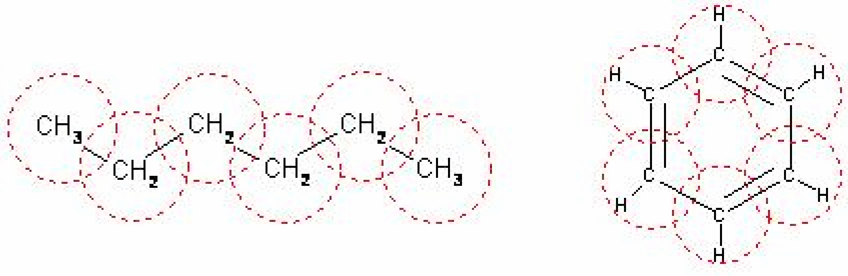
- parameterized to reproduce experimental data
- several popular force fields (ff) have been independently developed

Statistical force fields

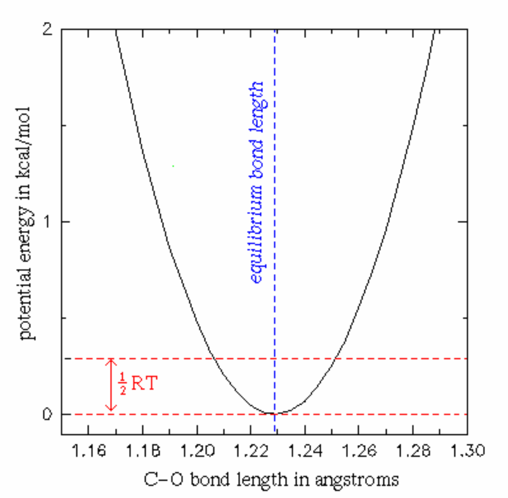
- does not correspond to a physical force but included to represent various statistical biases in nature
- crude model of interactions that are too difficult to represent accurately

Empirical force fields

- Amber, CHARMM, OPLS, GROMOS
- each ff models interactions using different functional forms
- may represent every atom separately ("**all atom**" representation) or only heavy atoms plus polarizable hydrogens ("**united atom**" representation)
- compromise between speed and accuracy—united atom is faster
- all have bonded and non-bonded terms



all atom vs. united atom representations



© O. S. Smart, 1995

CHARMM

- **C**hemistry at **HAR**vard **M**olecular **M**echanics
- Current release is for all-atom representation
- Different ff for proteins, carbohydrates, nucleic acids, lipids, etc
- In the most basic form, contains six terms to describe biomolecular structure

$$U(\vec{R}) = \sum_{bonds} K_b(b-b_0)^2 + \sum_{angles} K_\theta(\theta-\theta_0)^2$$

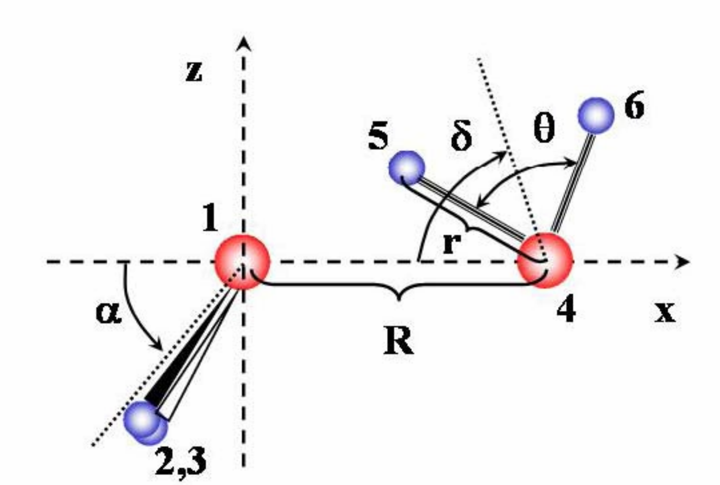
$$+ \sum_{dihedral} K_\chi\Big(1+\cos(n\chi-\delta)\Big) + \sum_{impropers} K_{imp}(\varphi-\varphi_0)^2$$

$$+ \sum_{nonbond}\left(\varepsilon_{ij}\left[\left(\frac{R\min_{ij}}{r_{ij}}\right)^{12} - \left(\frac{R\min_{ij}}{r_{ij}}\right)^{6}\right]\right) + \frac{q_i q_j}{\varepsilon r_{ij}}$$

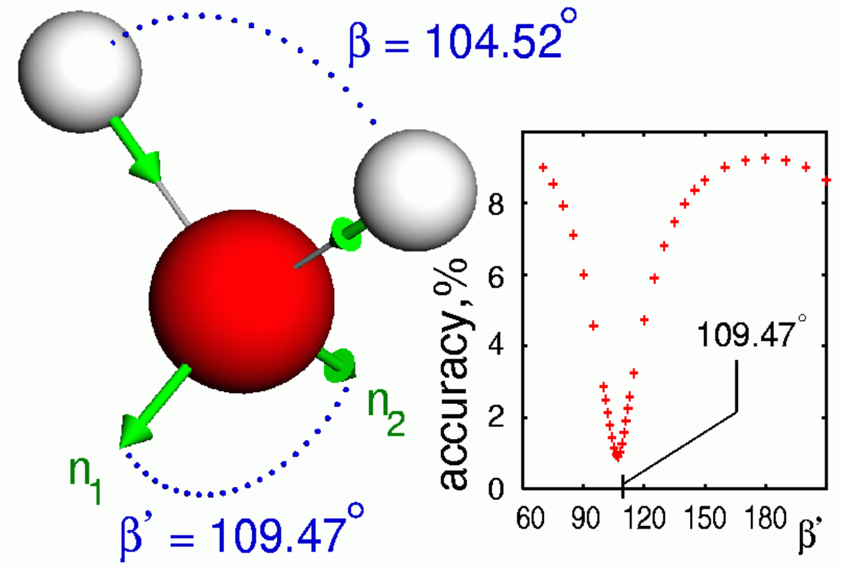
MacKerell, J Comp Chem 25, 1584 (2004)

Solvent treatment

- different models of water (e.g. TIP3P, TIP4P, SPC, extended SPC/E)
- TIP3P: electrostatic interaction truncated at 8.5 Å
- ff need to be developed with the solvent molecule in mind



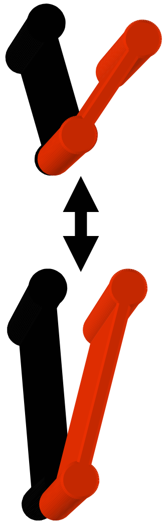
Model	SPC	SPC/E	TIP3P
r (Å)	1.0	1.0	0.9572
θ (°)	109.47	109.47	104.52
A / kcal Å^{12}/mol	629400	629400	582000
C / kcal Å^{6}/mol	625.5	625.5	595.0
Charge on oxygen	-0.82	-0.8472	-0.834



tip4p water model

$$U_{vdw} = \frac{A}{r^{12}} - \frac{C}{r^{6}}$$

Refinements



Lone pair electrons to improve hydrogen bond prediction

Coupling between internal coordinates

e.g. as angle decreases, the bond length increases

Anharmonic term to better reproduce vibrational spectra

Hyperconjugation and electronegativity

Modification of non-bonded Lennard-Jones term

e.g. 12-6 ➔ 9-6, addition of 14-7 term

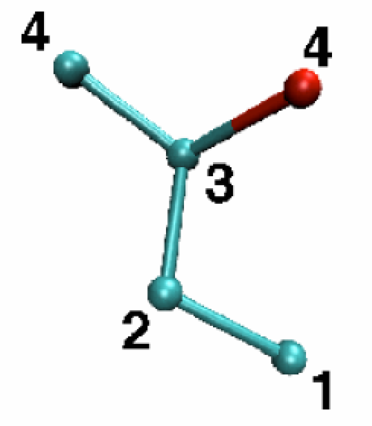
Polarizability of atoms

$$U_{pol} = -\sum \mu_i \bullet E_i$$

fluctuating charge in CHARMM to capture polarizability

off center charge in AMBER

1-4 interaction may be scaled or not



1-4 interaction

Statistical potential

Existing protein structures already contain a vast amount of information correlating sequence with structure

This information may be extracted by creating a “**knowledge-based potential (KBP)**”, i.e. a database-driven energy function, based on the frequency of different structural arrangements

KBP may complement empirical ff during protein design, protein folding, and ligand binding analysis

- Poole and Ranganathan, COSB 16, 508 (2006)
- Buchete et al, COSB 14, 225 (2004)
- Gohlke and Klebe, COSB 11, 231 (2001)

Users are allowed to introduce potential terms based on experience

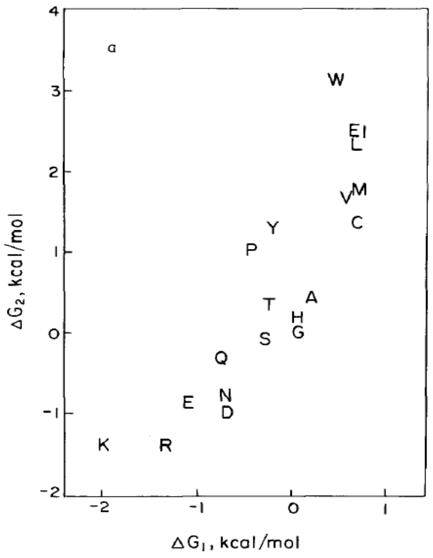
e.g. helix propensity, solvent exposure

Boltzmann hypothesis

The specific interaction in the database of known protein structures occurs with a frequency that depends on its free energy according to the Boltzmann distribution

probability(y) ~ exp(- free energy (y) * constant)

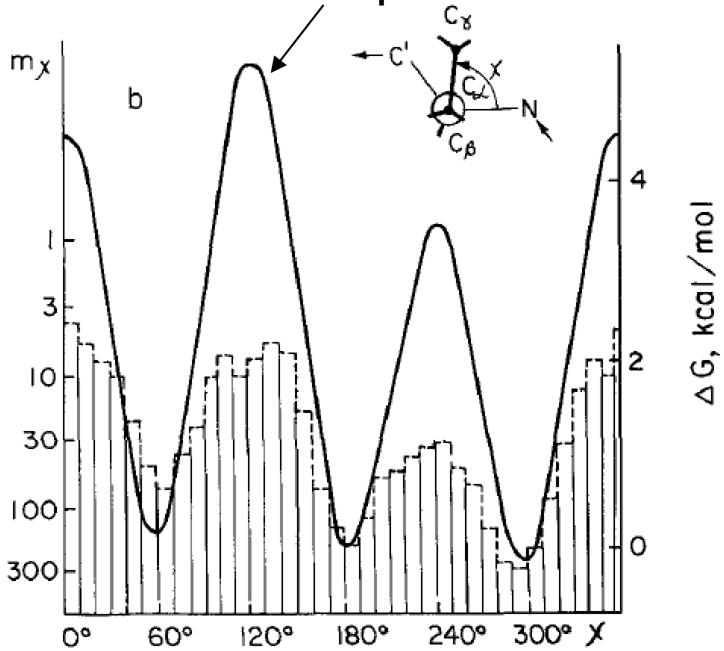
organic/water transfer energy



$$\Delta G_1 = -RT\log(f)$$

$$f = \frac{N_s / \sum N_s}{N_b / \sum N_b}$$

computed



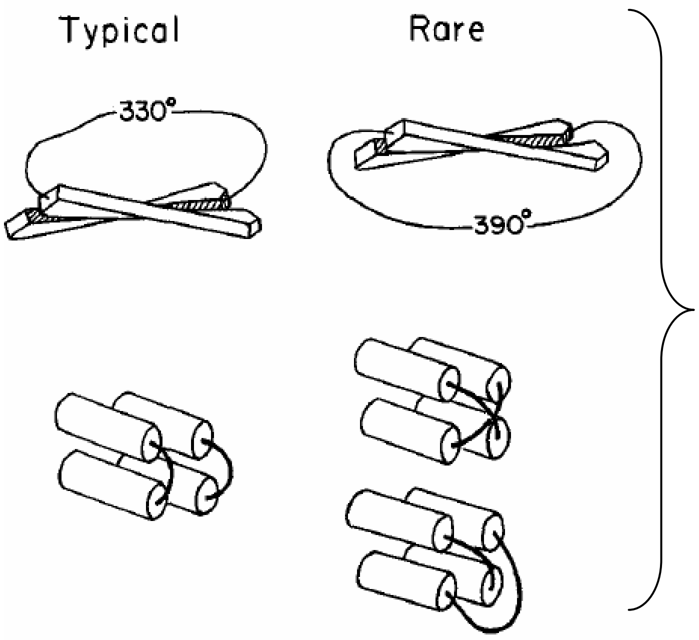
$$\Delta G_1 = -RT\log(\text{no. of side chains})$$

Finkelstein et al,
Proteins 23, 142 (1995)

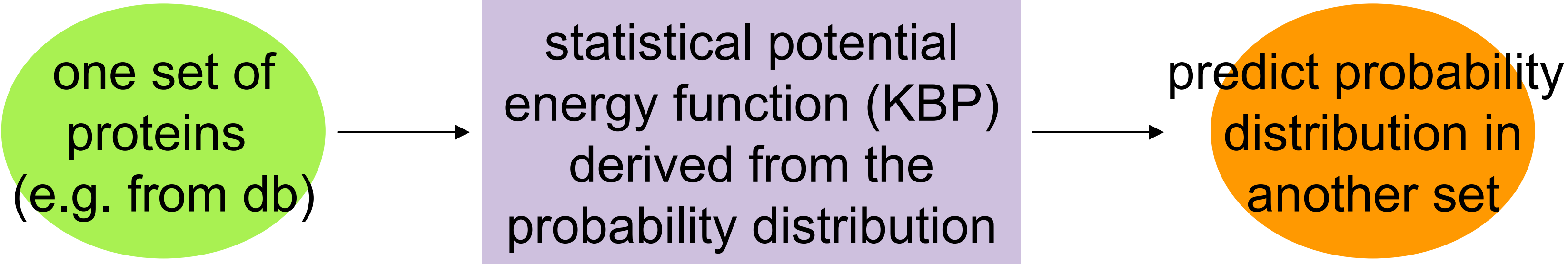
Extracting and applying KBP

Structural features that may obey the Boltzmann hypothesis include:

hydrogen bonds, hydrophobicity, proline isomerization, internal cavities, side chain-side chain interactions, and interactions at the level of specific atom types



➔ create a potential to favor the more commonly observed conformations



one set of proteins (e.g. from db)

→

statistical potential energy function (KBP) derived from the probability distribution

→

predict probability distribution in another set

Rotamer

Different side chain conformations are not found in equal distribution over the dihedral angle space but tend to cluster at specific regions of the space

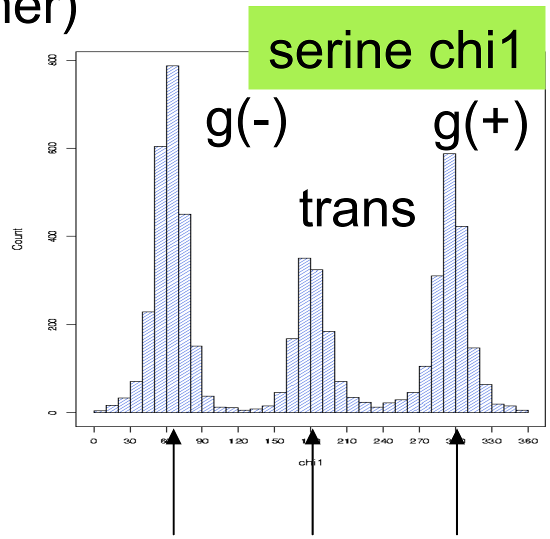
The set of side chain dihedral angles corresponding to a local energy minimum is called a rotamer (short for rotational isomer)

e.g. one rotamer of Leu = (60°, -60°)

non-rotamer of Leu = (60°, 0°)

Ala has one rotamer

Lys has 81 rotamers



Bond length and angle are presumed fixed

Discretization of the side chain dihedral degrees of freedom simplifies the sequence search—standard practice for computational protein design

Rotamer library

Rotamer library is a collection of all rotamers of all amino acids

contains dihedral angle information and probability of each

May or may not depend on the backbone conformation

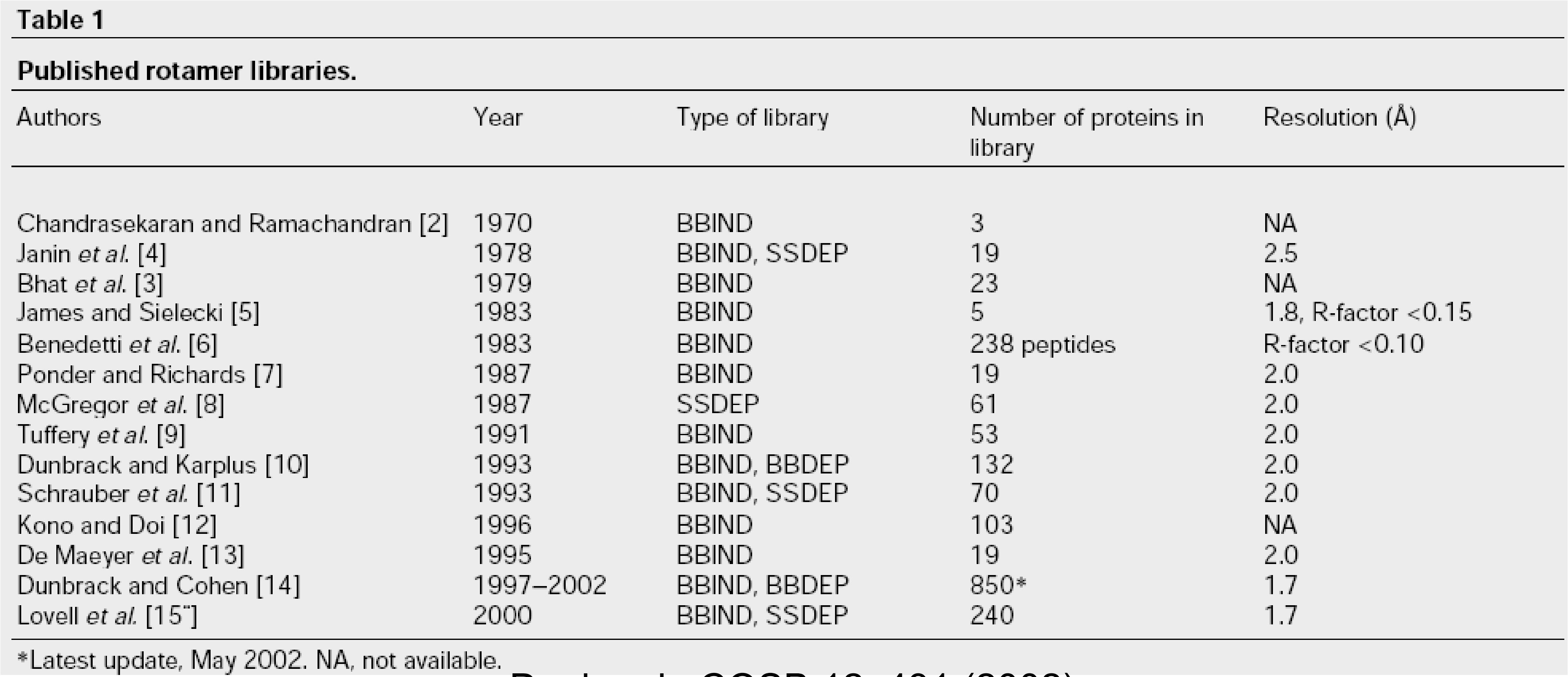


Table 1

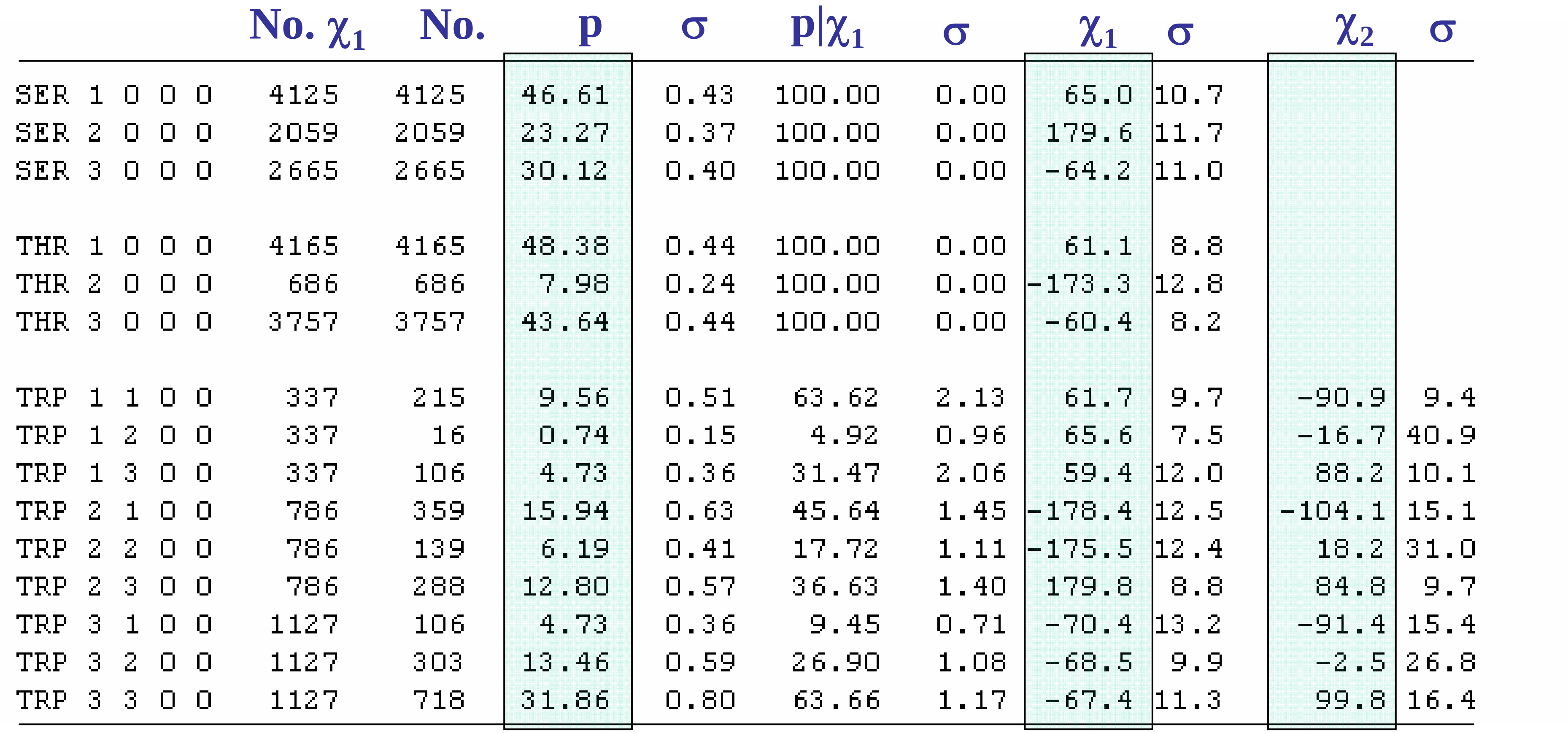
Published rotamer libraries.

Authors	Year	Type of library	Number of proteins in library	Resolution (Å)
Chandrasekaran and Ramachandran [2]	1970	BBIND	3	NA
Janin *et al.* [4]	1978	BBIND, SSDEP	19	2.5
Bhat *et al.* [3]	1979	BBIND	23	NA
James and Sielecki [5]	1983	BBIND	5	1.8, R-factor <0.15
Benedetti *et al.* [6]	1983	BBIND	238 peptides	R-factor <0.10
Ponder and Richards [7]	1987	BBIND	19	2.0
McGregor *et al.* [8]	1987	SSDEP	61	2.0
Tuffery *et al.* [9]	1991	BBIND	53	2.0
Dunbrack and Karplus [10]	1993	BBIND, BBDEP	132	2.0
Schrauber *et al.* [11]	1993	BBIND, SSDEP	70	2.0
Kono and Doi [12]	1996	BBIND	103	NA
De Maeyer *et al.* [13]	1995	BBIND	19	2.0
Dunbrack and Cohen [14]	1997–2002	BBIND, BBDEP	850*	1.7
Lovell *et al.* [15••]	2000	BBIND, SSDEP	240	1.7

*Latest update, May 2002. NA, not available.

Dunbrack, COSB 12, 431 (2002)

Backbone independent library



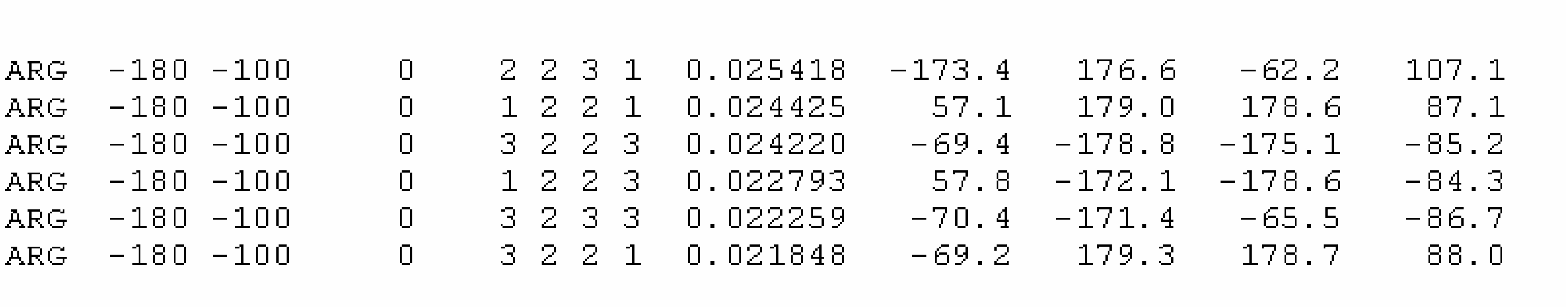
					No. χ_1	No.	p	σ	p|χ_1	σ	χ_1	σ	χ_2	σ
SER	1	0	0	0	4125	4125	46.61	0.43	100.00	0.00	65.0	10.7		
SER	2	0	0	0	2059	2059	23.27	0.37	100.00	0.00	179.6	11.7		
SER	3	0	0	0	2665	2665	30.12	0.40	100.00	0.00	-64.2	11.0		
THR	1	0	0	0	4165	4165	48.38	0.44	100.00	0.00	61.1	8.8		
THR	2	0	0	0	686	686	7.98	0.24	100.00	0.00	-173.3	12.8		
THR	3	0	0	0	3757	3757	43.64	0.44	100.00	0.00	-60.4	8.2		
TRP	1	1	0	0	337	215	9.56	0.51	63.62	2.13	61.7	9.7	-90.9	9.4
TRP	1	2	0	0	337	16	0.74	0.15	4.92	0.96	65.6	7.5	-16.7	40.9
TRP	1	3	0	0	337	106	4.73	0.36	31.47	2.06	59.4	12.0	88.2	10.1
TRP	2	1	0	0	786	359	15.94	0.63	45.64	1.45	-178.4	12.5	-104.1	15.1
TRP	2	2	0	0	786	139	6.19	0.41	17.72	1.11	-175.5	12.4	18.2	31.0
TRP	2	3	0	0	786	288	12.80	0.57	36.63	1.40	179.8	8.8	84.8	9.7
TRP	3	1	0	0	1127	106	4.73	0.36	9.45	0.71	-70.4	13.2	-91.4	15.4
TRP	3	2	0	0	1127	303	13.46	0.59	26.90	1.08	-68.5	9.9	-2.5	26.8
TRP	3	3	0	0	1127	718	31.86	0.80	63.66	1.17	-67.4	11.3	99.8	16.4

Lovell rotamer library

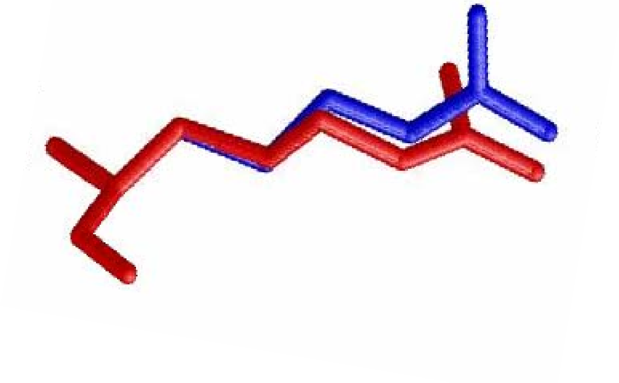
- backbone independent
- http://kinemage.biochem.duke.edu/databases/rotamer.html

Backbone dependent library

Side chain rotameric probabilities are compiled separately for each phi/psi main chain angle pairs



	ϕ	φ			p	χ_1	χ_2	χ_3	χ_4
ARG	-180	-100	0	2 2 3 1	0.025418	-173.4	176.6	-62.2	107.1
ARG	-180	-100	0	1 2 2 1	0.024425	57.1	179.0	178.6	87.1
ARG	-180	-100	0	3 2 2 3	0.024220	-69.4	-178.8	-175.1	-85.2
ARG	-180	-100	0	1 2 2 3	0.022793	57.8	-172.1	-178.6	-84.3
ARG	-180	-100	0	3 2 3 3	0.022259	-70.4	-171.4	-65.5	-86.7
ARG	-180	-100	0	3 2 2 1	0.021848	-69.2	179.3	178.7	88.0



dihedral angles are not always as we might expect them to be—true for both backbone-independent and backbone-dependent libraries

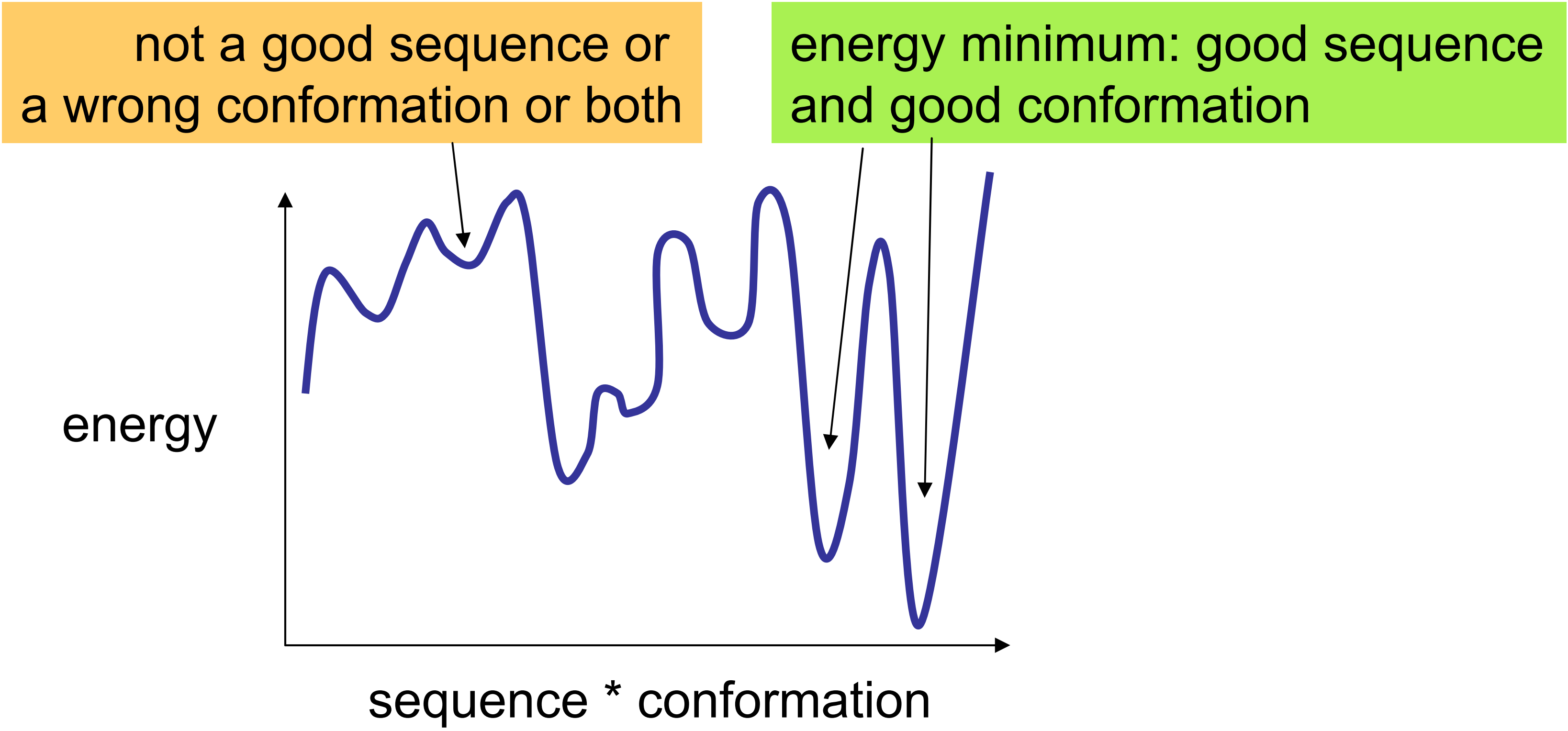
Dunbrack rotamer libraries

- backbone-independent and backbone-dependent
- http://dunbrack.fccc.edu/bbdep/bbdepdownload.php

Search algorithms

CPD requires evaluating the compatibility of a sequence with the target structure by calculating the interaction energy based on a force field

For each proposed sequence, all possible side chain conformations must be considered



not a good sequence or a wrong conformation or both

energy minimum: good sequence and good conformation

Stochastic v. deterministic

Stochastic search traces a search path that is inherently random

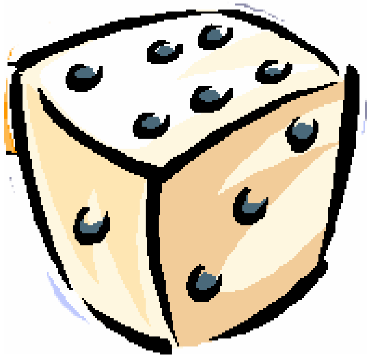
- Each time the search is conducted, it'll proceed through a different sequence of events
- Relies on a large number of queries to identify the minimum energy configuration

Types of stochastic search used in protein design

- Monte Carlo, Monte Carlo with Metropolis cut
- Simulated annealing
- Genetic and evolutionary algorithms
- Tabu search

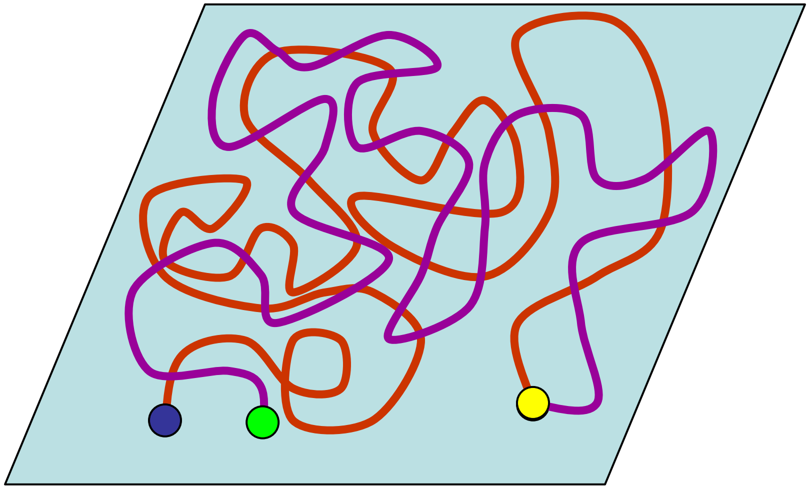
Monte Carlo search

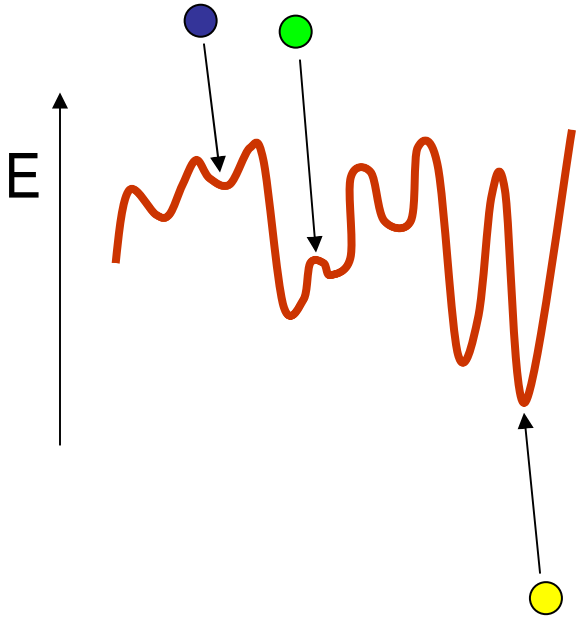
- Initialize the system by assigning an amino acid to each residue position and also choosing a rotamer state for the amino acid
- Choose the next potential configuration at random (e.g. picking a random number and mapping it to a new sequence and/or conformation)
- Accept the move and re-initialize the system or reject the move based on whether the new proposed configuration has a lower energy
- Systematically lower the total energy
- There are virtually an infinite number of different ways of lower the energy
- After a while, the number of failed attempts between successful tries may increase dramatically, making the search highly inefficient



Search the phase space

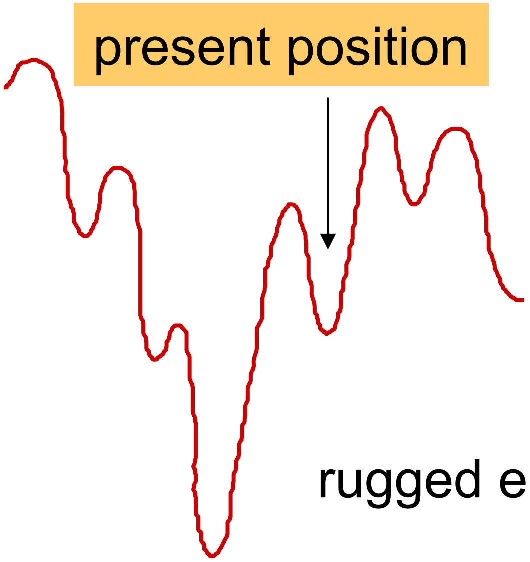
Ideally, we would like to find the minimum energy configuration regardless of the initial conditions





Getting trapped in local minima

Search may come to a grinding halt after getting stuck in a local minimum



rugged energy landscape

whatever you do, you end up raising the energy of the system, so the search never moves forward from the present position

<u>Solution</u>

Accept the moves according to the "**Metropolis**" rule

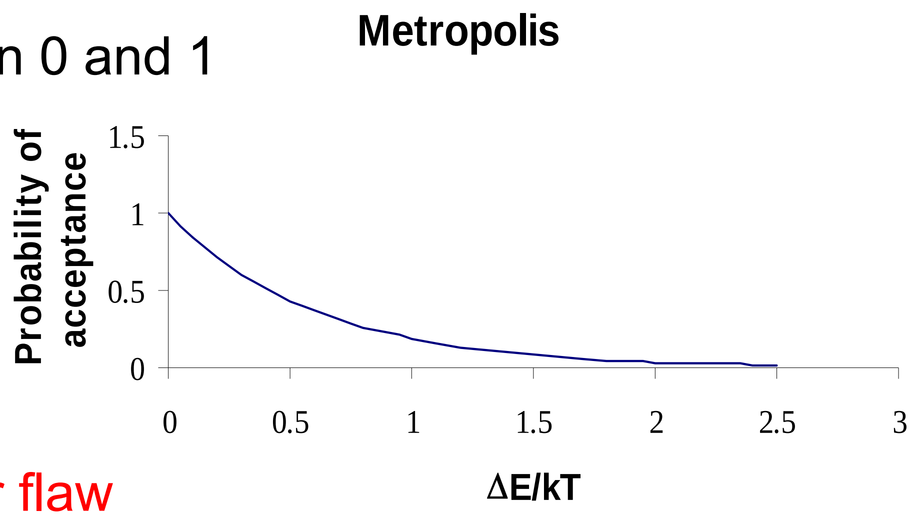
If $\Delta E < 0$ accept the move—as before

If $\Delta E > 0$ also accept the move but with a probability $p = e^{\frac{-\Delta E}{K_B T}}$

Pick a random real number *x* between 0 and 1

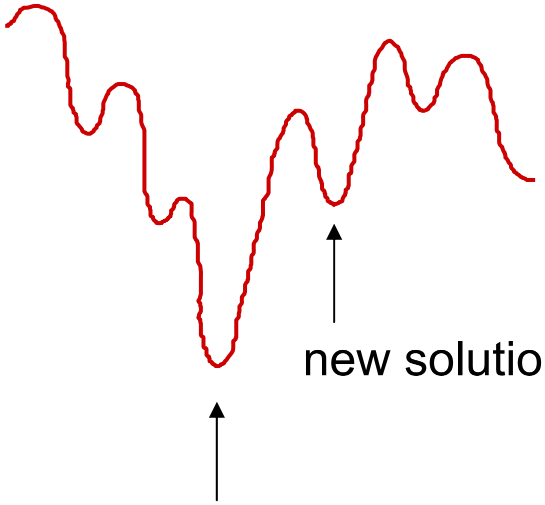
If

$$x < \exp\left(-\frac{\Delta E}{k_B T}\right)$$



This modified search still has a major flaw

Applying the Metropolis criteria can spoil the search by abandoning a good solution that was difficult to find in the first place (i.e. took many tries to get to this solution) for a suboptimum solution



new solution that was accepted by the Metropolis criteria

present solution

Simulated annealing

Annealing is a process in which the microstructure of a material is altered by heating followed by a gradual cooling in order to change its strength and hardness

Gradual cooling allows individual atoms to avoid internal stress and seek out the ground energy state

Simulated annealing (SA) starts by "heating" the system under investigation and gradually lowers the temperature, while resolving internal conflicts through evaluation of an appropriate metric

$$\text{acceptance probability} = \exp\left(-\frac{\Delta E}{k_B T}\right)$$

Applications of SA

- Traveling salesman problem and other *NP*-hard optimization problems
- Controlling the movement of a robotic arm to move a glass
- Pinpointing the position of a sniper by the supersonic boom created by a bullet

Kirkpatrick et al, Science 220, 671 (1983)

Protein design by SA

Protocol

1. Initialize the sequence and conformation as in Monte Carlo
2. Randomly suggest a change in sequence and/or conformation
3. The new “solution” is accepted according to the Metropolis criteria
4. The temperature in the Metropolis cutoff (i.e. $\exp(-\Delta E/kT)$) is gradually lowered, which has a consequence of making it more difficult to accept an energetically unfavorable move at a later stage of design

Designing protein by SA depends critically on

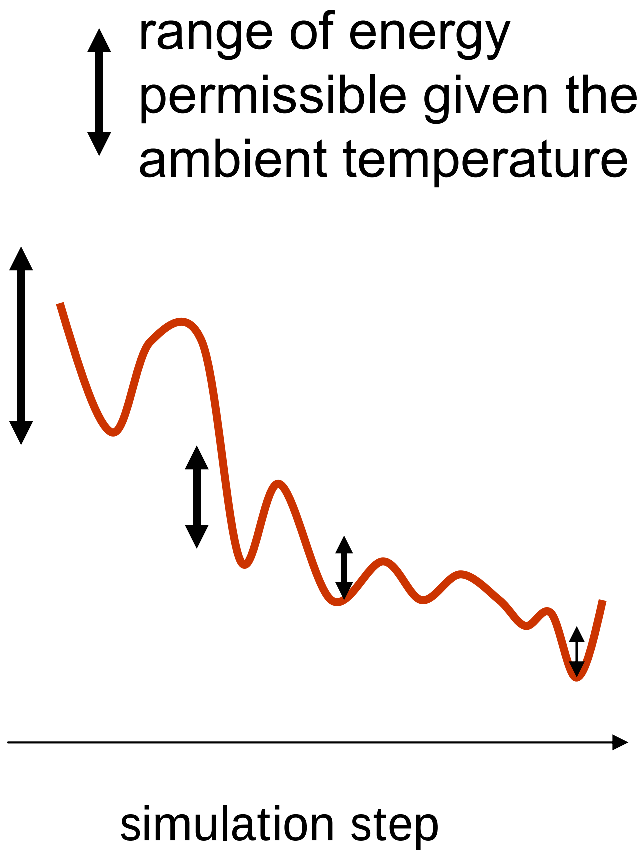
- energy function
- cooling schedule

Commonly used cooling schedule : $kT \sim 1/N$, where N is # of simul. step

Energy function : combination of empirical and statistical potential

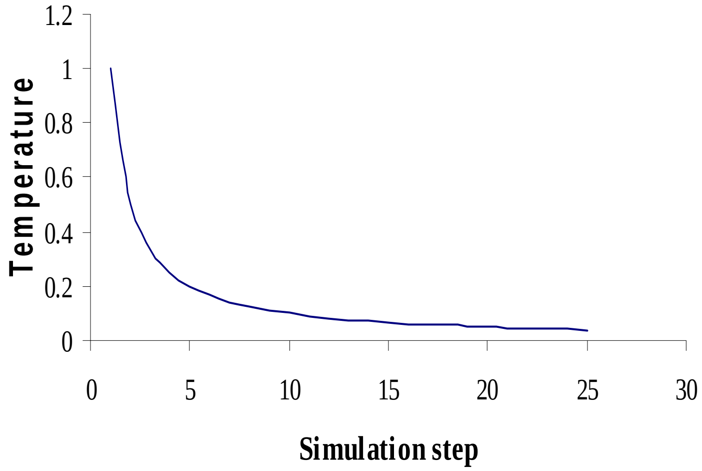
Setting the initial temperature high allows different parts of the phase space to be efficiently sampled

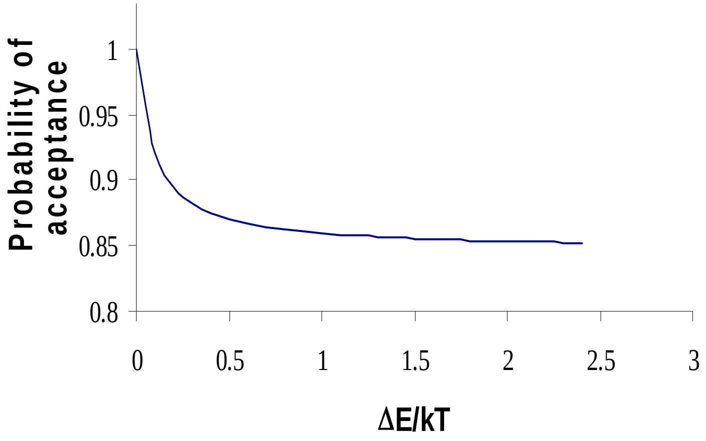
Progressively lowering the temperature prevents a good solution from being accidentally discarded as in the standard Monte Carlo simulation



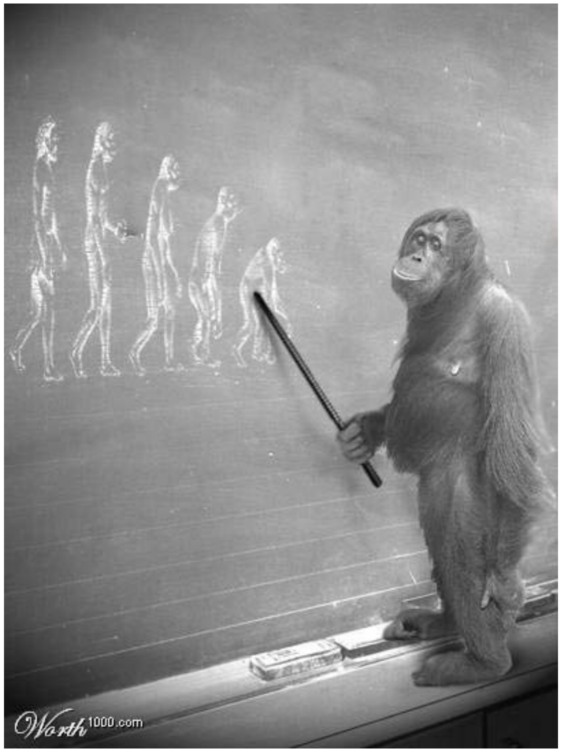
range of energy permissible given the ambient temperature

simulation step





Genetic and evolutionary algorithms



Theory of evolution according to Darwin is based on the survival of the fittest

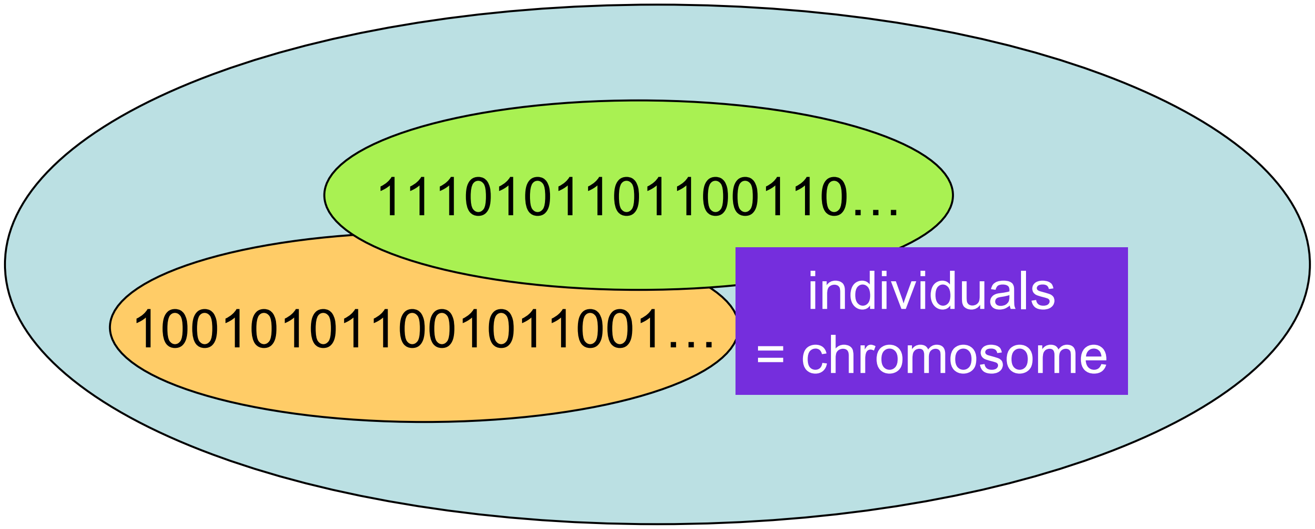
Essential elements of Darwinian evolution

- mechanism of diversity generation
  - random mutation
  - genetic recombination
- selection based on fitness criteria
- reproduction

Genetic algorithms (GA) and evolutionary algorithms (EA) are computer simulations of the evolutionary process in order to optimize an arbitrary fitness function, e.g. energy function

Nomenclature

- Collection of individuals: population
- Individuals and the genetic information they carry: genes (real life) and chromosomes (GA/EA)
  - chromosomes represent potential solutions
- Representation of information: string of integers (binary or not) or a real number
- Fitness evaluation: energy function—problem-specific



entire population

Representation of sequence in GA

- Represent the proposed amino acid sequence as:

  string of 1’s and 0’: 1001010101100110…

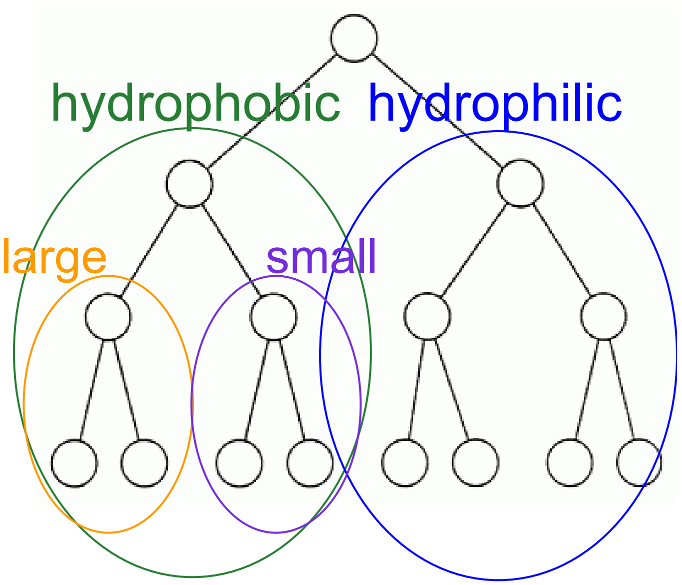
  there are 20 amino acids and ~200 rotamers per position

  use 8 bits per position

  $$\underbrace{10010101}_{\text{residue 1}}\ \underbrace{01100110}_{\text{residue 2}}\ldots$$

  some amino acids are over-represented

  but it may be possible to encode “closeness” by grouping amino acids on a binary tree



- Alternatively, represent the sequence as:

  S_1-S_2- … - S_m

  where S_i is the amino acid at the *i*-th position (as usual)

- Make multiple copies of these “chromosomes”—typically ~ 100 copies

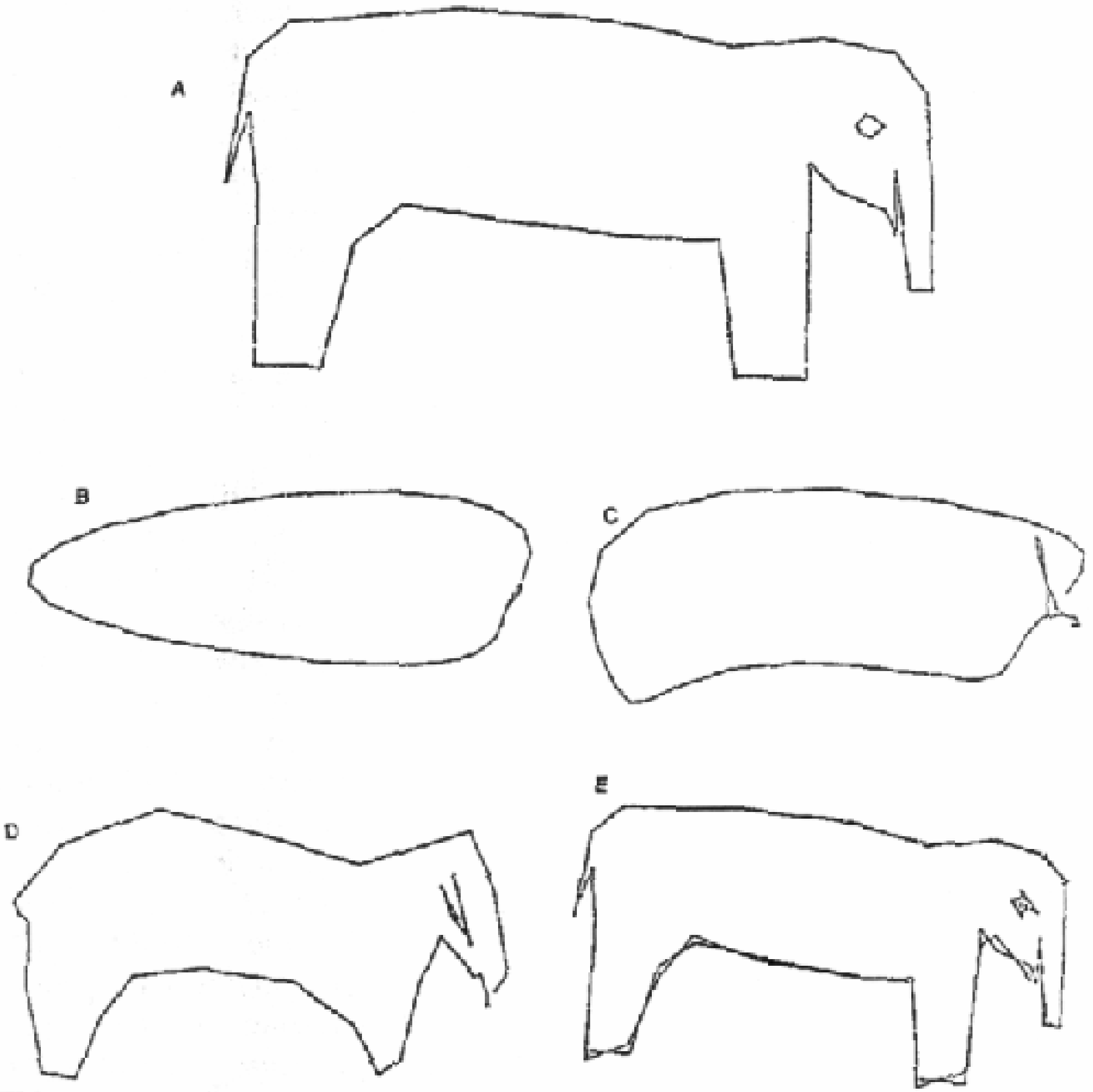


FIGURE 1.2. "How many parameters does does it take to fit an elephant?" was answered by Wel (1975). He started with an idealized drawing (A) defined by 36 points and used least squares Fourier sine series fits of the form $x(t) = \alpha_0 + \sum \alpha_i \sin(it\pi/36)$ and $y(t) = \beta_0 + \sum \beta_i \sin(it\pi/36)$ for $i = 1, \ldots, N$. He examined fits for $K = 5, 10, 20$, and 30 (shown in B–E) and stopped with the fit of a 30 term model. He concluded that the 30-term model "may not satisfy the third-grade art teacher, but would carry most chemical engineers into preliminary design."

Schema

Evaluate the “fitness” of each chromosome (i.e. sequence)

if the interaction energy is used as a metric, a sequence with lower energy (i.e. more stable) has a higher fitness

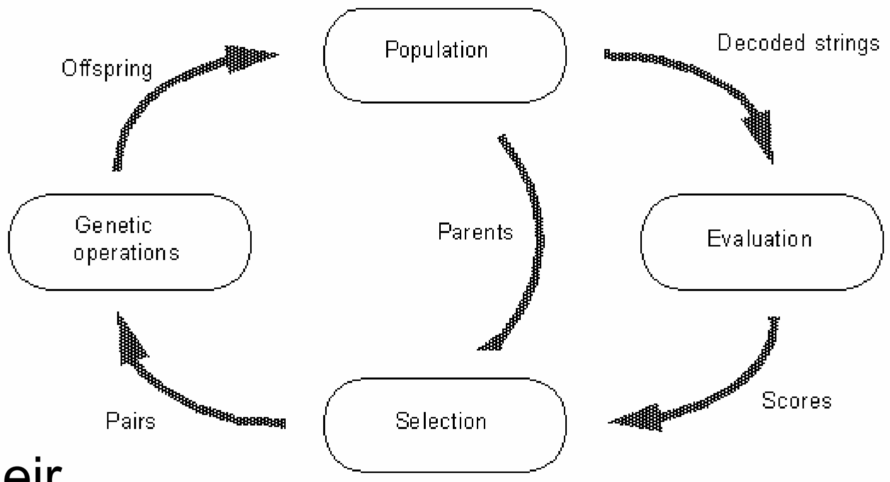

Population
Decoded strings
Evaluation
Scores
Selection
Pairs
Genetic operations
Offspring
Parents

Figure 5.2: The “reproduction” cycle.

Rank order chromosomes based on their fitness

chromosomes that are ranked higher will have a better chance of “survival”

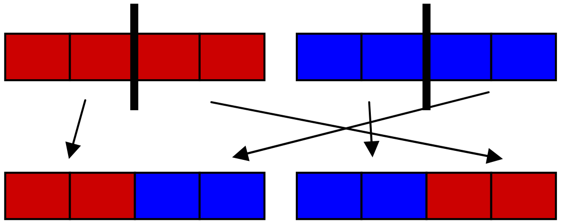
Mutate and recombine (described next)

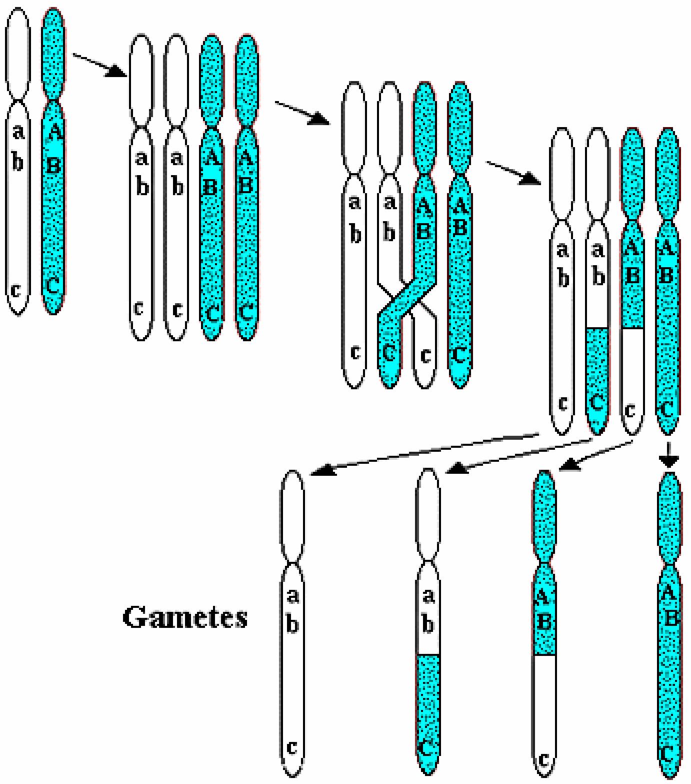
Replace the least fit individuals with these next-generation chromosomes

Diversity generation

Cross-over (recombination)

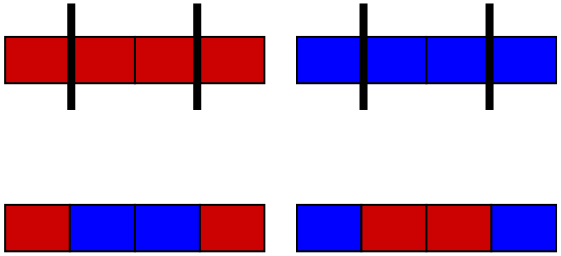
- pick two chromosomes from near the top of the ranking (e.g. top 20%)
- pick a cross-over point
- swap the string to the right of the cross-over point
- need to optimize the rate of recombination



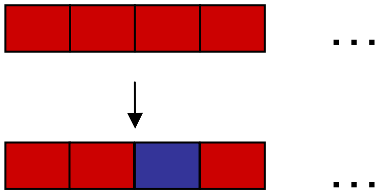


Crossing-over and recombination during meiosis

Multiple points cross-over is also possible but the advantage is not clear



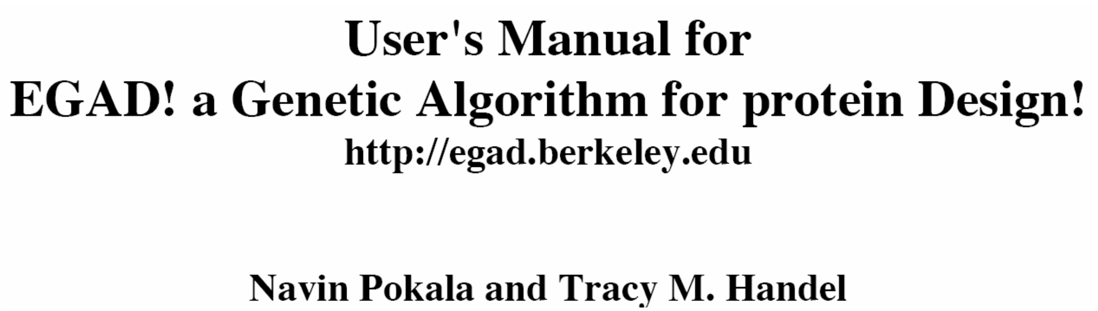
Mutation : for each chromosome, mutate the value at a randomly selected position to something else



Optimizing the rate of mutation is non-trivial

- low rate of mutation may slow the search process too much
- high rate of mutation may ruin successful search
- possible solution: lower the rate of mutation with iteration
- e.g. 1% mutation rate means an average of 1 mutation per 100 amino acids for each chromosome

Downloadable, GA-based protein design algorithm



User's Manual for
EGAD! a Genetic Algorithm for protein Design!
http://egad.berkeley.edu

Navin Pokala and Tracy M. Handel

(Self-consistent) mean field theory

The number of potential solutions is much too large to enumerate

Avoid combinatorial explosion by replacing all interactions to any one body with an average interaction from the rest—essentially converts a many-body problem to a one-body problem

Use of mean field theory (MFT) in protein engineering

- conformational optimization
- structure prediction on a lattice
- loop construction in protein homology modeling
- sequence design
- side chain modeling

Mean field energy

$$E(r_1,\ldots,r_N)=\sum_i p_i(r_i)\bullet\xi_i(r_i)+\frac{1}{2}\bullet\sum_i\sum_{j\neq i}p_i(r_i)p_j(r_j)\varepsilon_{i,j}(r_i,r_j)$$

$p_i(r_i)$ is the probability of r_i, i.e. residue i in state r

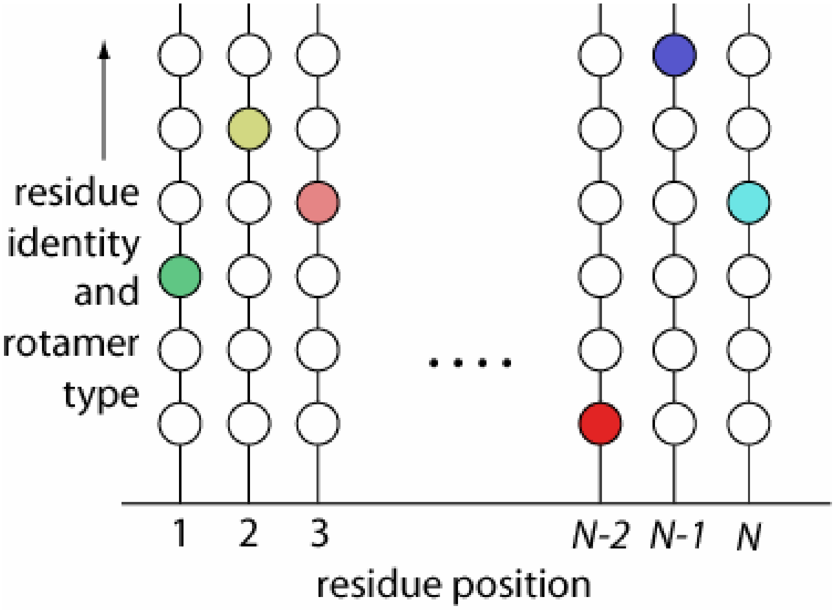
(combination of residue identity and rotamer state)

$\xi_i(r_i)$ is the internal energy of r_i

$\varepsilon_{i,j}(r_i,r_j)$ is the interaction energy between r_i and r_j

The probability of *i*-th residue being in the state "*r*" is computed from

$$p_i(r_i)=\frac{\exp\left(-\frac{W(r_i)}{kT}\right)}{\sum_{r_i}\exp\left(-\frac{W(r_i)}{kT}\right)}$$



scads

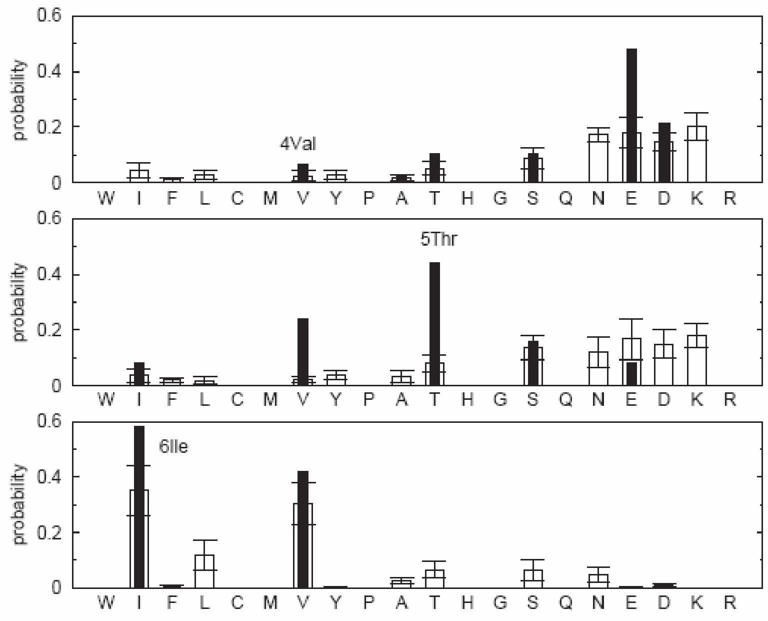
Statistical, computation assisted design strategy – J. Saven (U Penn)

Implementation of mean field calculation

Compute the probability distribution of residues at each randomized position by maximizing a quantity that looks like informational entropy

$$S = -\sum_{\text{protein states}} p_{aa} \log(p_{aa})$$

protein states = residue identity + rotameric state


probability
4Val
5Thr
6Ile
W I F L C M V Y P A T H G S Q N E D K R