Small Molecule & Protein Docking

CHEM 430

Molecular Docking Models

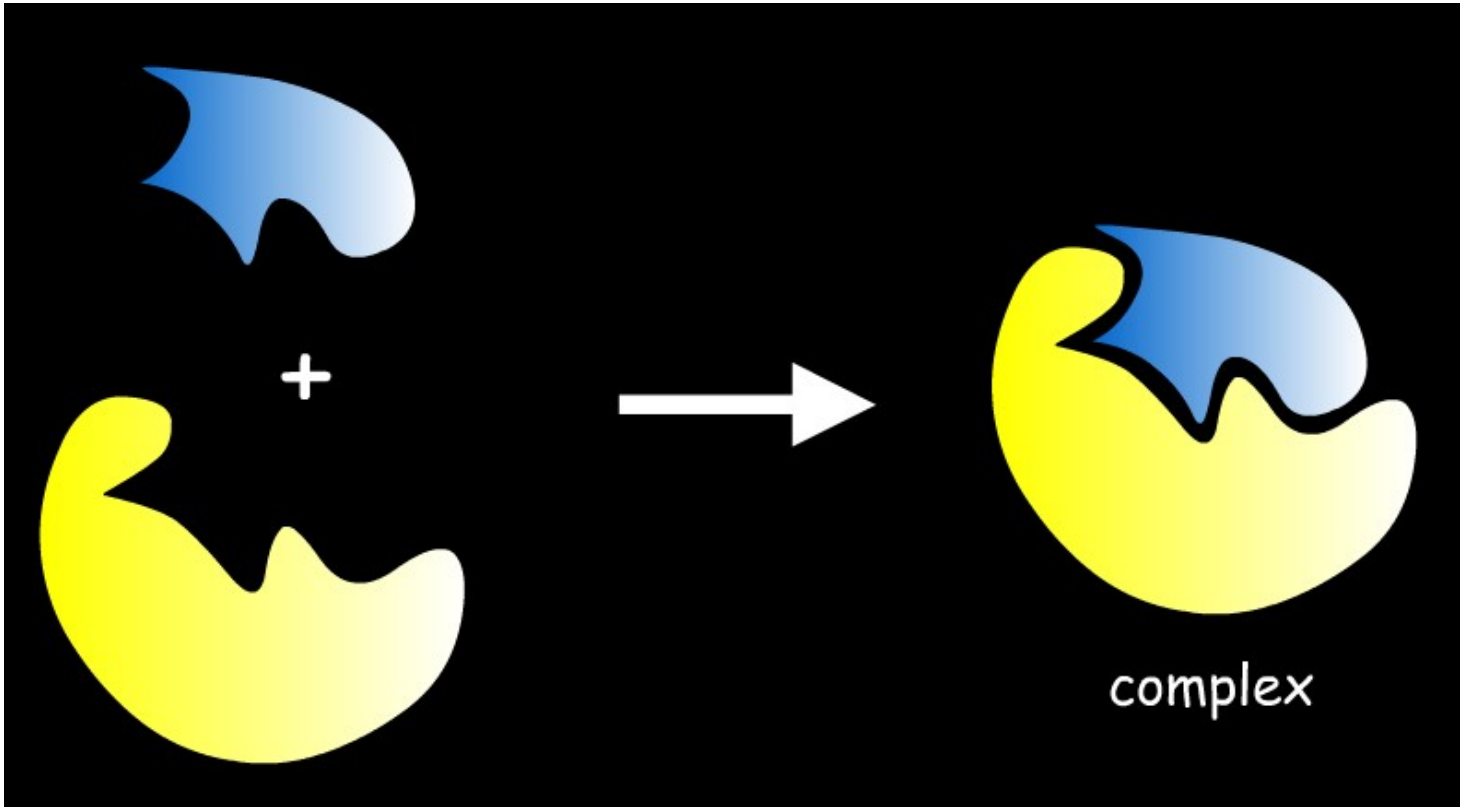
- Over the years biochemists have developed numerous models to capture the key elements of the molecular recognition process. Although very simplified, these models have proven highly useful to the scientific community.

year	model	author(s)
1890	lock-and-key	Emil Fischer
1958	induced-fit	Daniel Koshland
2003	conformation ensemble	Buyong Ma et al.

" All models are wrong, some are useful " (George Box)

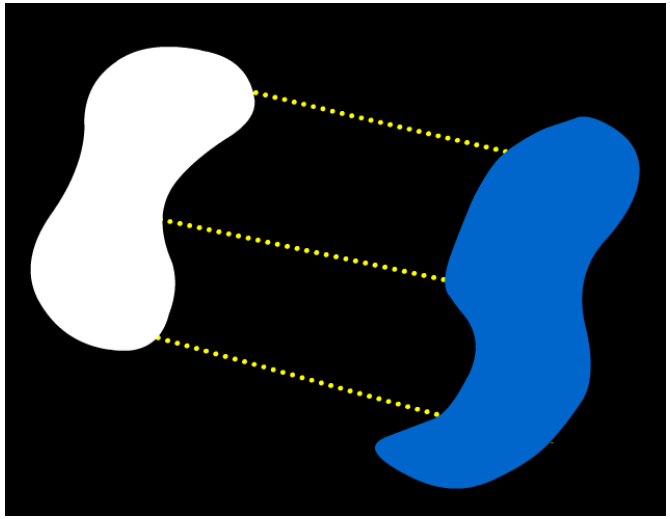
The Lock and Key Theory

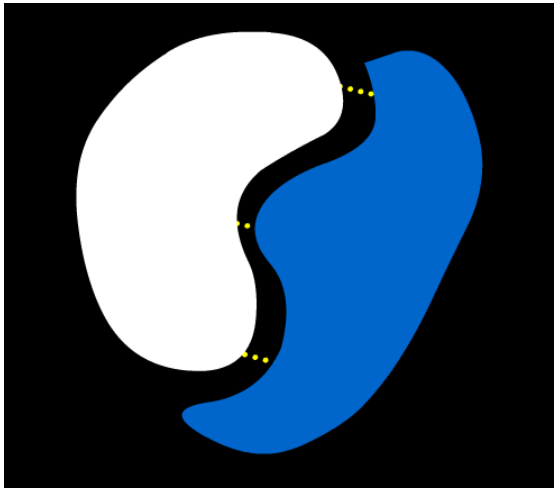
- As far back as 1890 Emil Fischer proposed a model called the "lock-and-key model" that explained how biological systems function. A substrate fits into the active site of a macromolecule, just like a key fits into a lock. Biological 'locks' have unique stereochemical features that are necessary to their function.



The Induced-Fit Theory

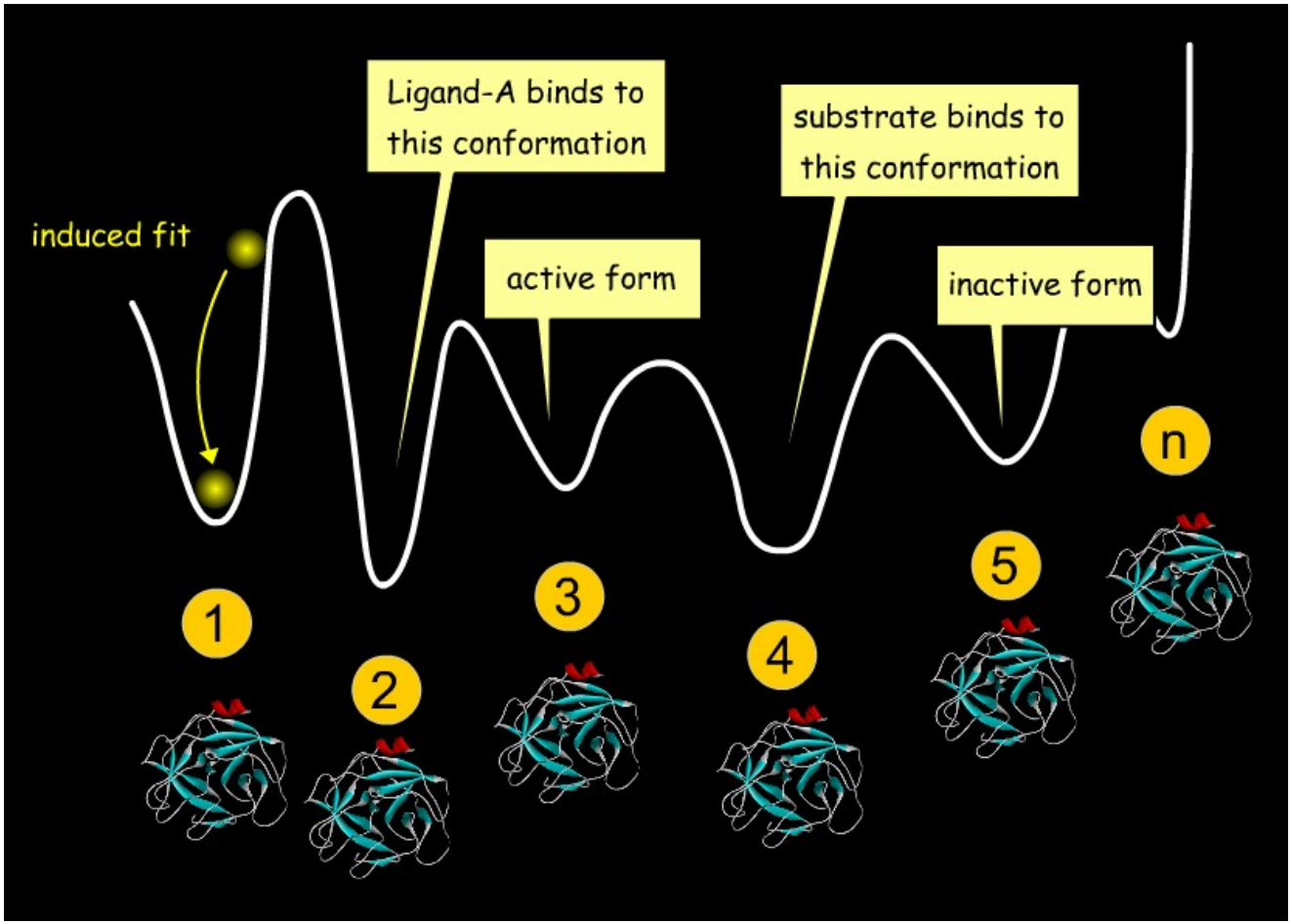
- In 1958 Daniel Koshland introduced the "induced-fit theory". The basic idea is that in the recognition process, both ligand and target mutually adapt to each other through small conformational changes, until an optimal fit is achieved.





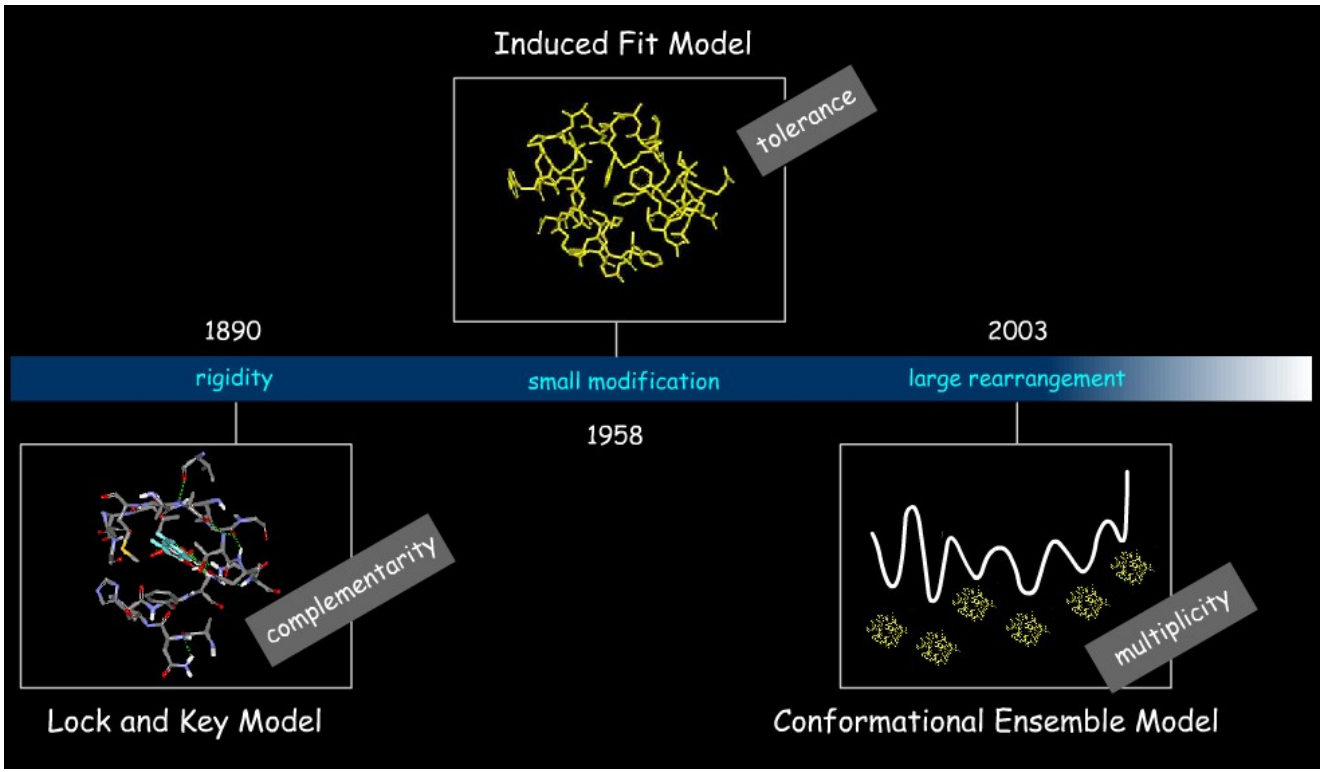
The Conformation Ensemble Model

- In addition to small induced-fit adaptation, it has been observed that proteins can undergo much larger conformational changes. A recent model describes proteins as a pre-existing ensemble of conformational states. The plasticity of the protein allows it to switch from one state to another.

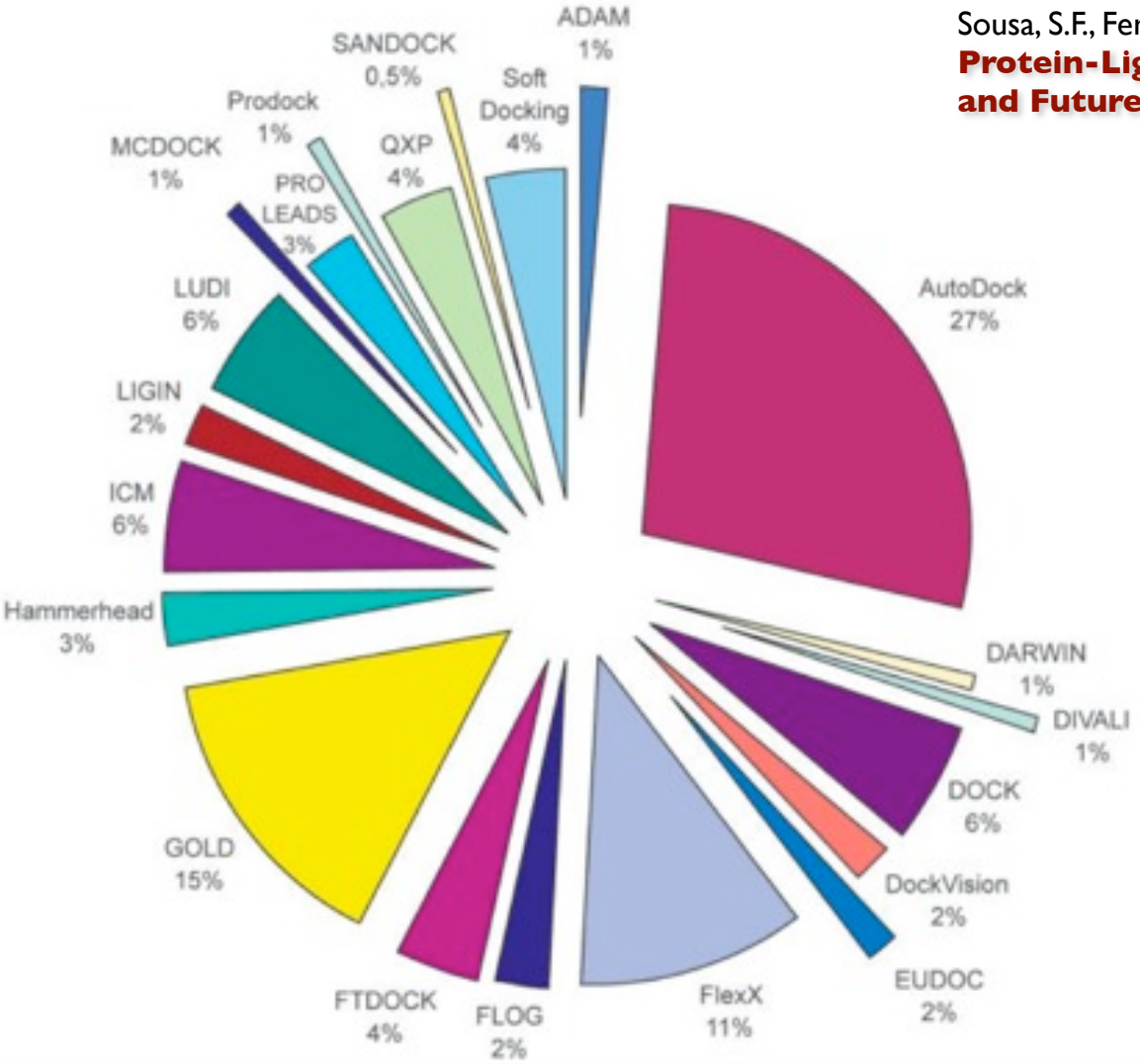


From the Lock and Key to the Ensemble Model

- Lock-and-key, induced-fit and the conformation ensemble model are not contradictory. Each one focuses on a particular aspect of the recognition process. The lock-and-key model introduces the principle of 3D complementarity, the induced-fit model explains how complementarity is achieved, and the ensemble model depicts the conformational complexity of proteins.



Number of Citations for Docking Programs —ISI Web of Science (2005)



Sousa, S.F., Fernandes, P.A. & Ramos, M.J. (2006) **Protein-Ligand Docking: Current Status and Future Challenges** *Proteins*, **65**:15-26

Introduction

- A significant portion of biology is built on the paradigm

  sequence → structure → function

- As we sequence more genomes and get more structural information, the next challenge will be to predict interactions and binding for two or more biomolecules (nucleic acids, proteins, peptides, drugs or other small molecules)

Introduction

- The questions we are interested in are:
  - Do two biomolecules bind each other?
  - If so, how do they bind?
  - What is the binding free energy or affinity?
- The goals we have are:
  - Searching for lead compounds
  - Estimating effect of modifications
  - General understanding of binding
  - …

Rationale

- The ability to predict the binding site and binding affinity of a drug or compound is immensely valuable in the area or pharmaceutical design
- Most (if not all) drug companies use computational methods as one of the first methods of screening or development
- Computer-aided drug design is a more daunting task, but there are several examples of drugs developed with a significant contribution from computational methods

Examples

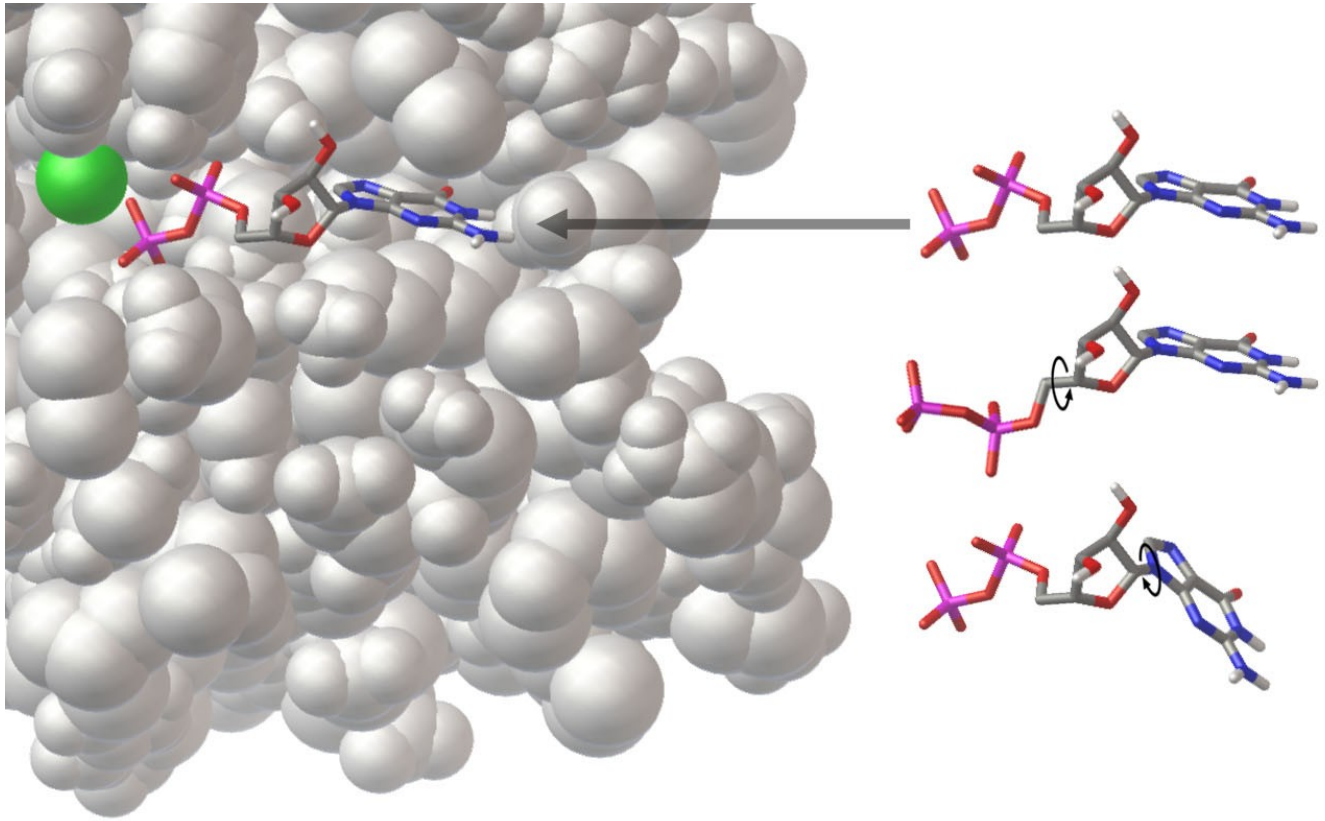
- Tacrine – inhibits acetylcholinesterase and boost acetylcholine levels (for treating Alzheimer's disease)
- Relenza – targets influenza
- Invirase, Norvir, Crixivan – Various HIV protease inhibitors
- Celebrex – inhibits Cox-2 enzyme which causes inflammation (not our fault)

Docking

- *Docking* refers to a computational scheme that tries to find the best binding orientation between two biomolecules where the starting point is the atomic coordinates of the two molecules

- Additional data may be provided (biochemical, mutational, conservation, etc.) and this can significantly improve the performance, however this extra information is not required

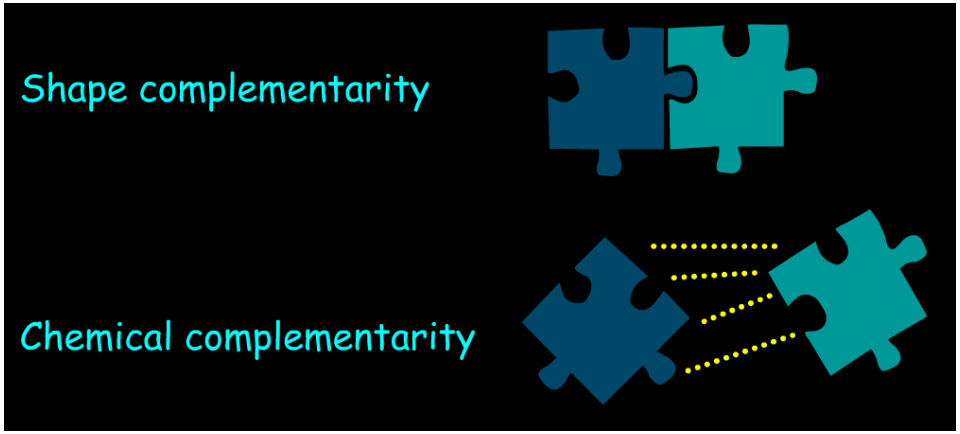
What is Docking?

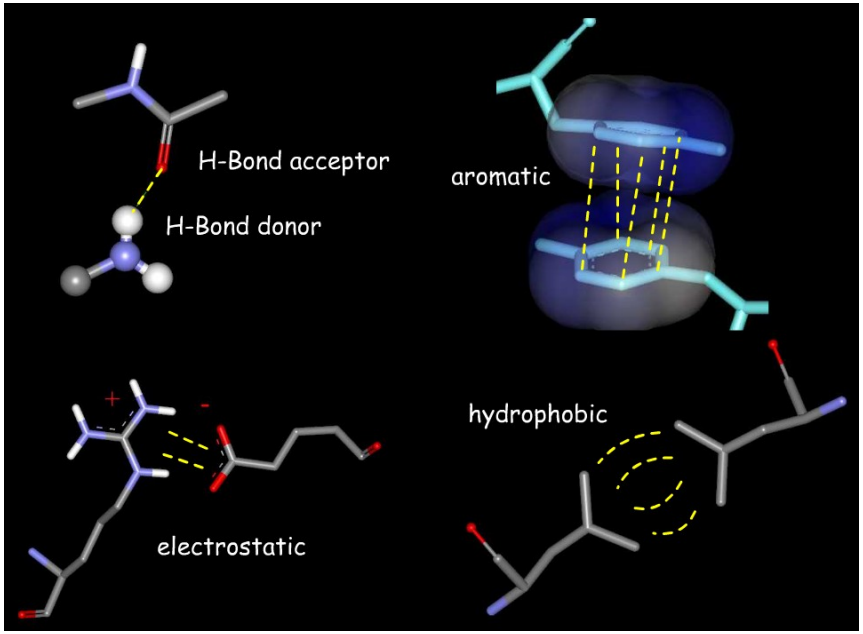


Given the 3D structures of two molecules, determine the best binding modes.

Molecular Complementarity in Computational Docking

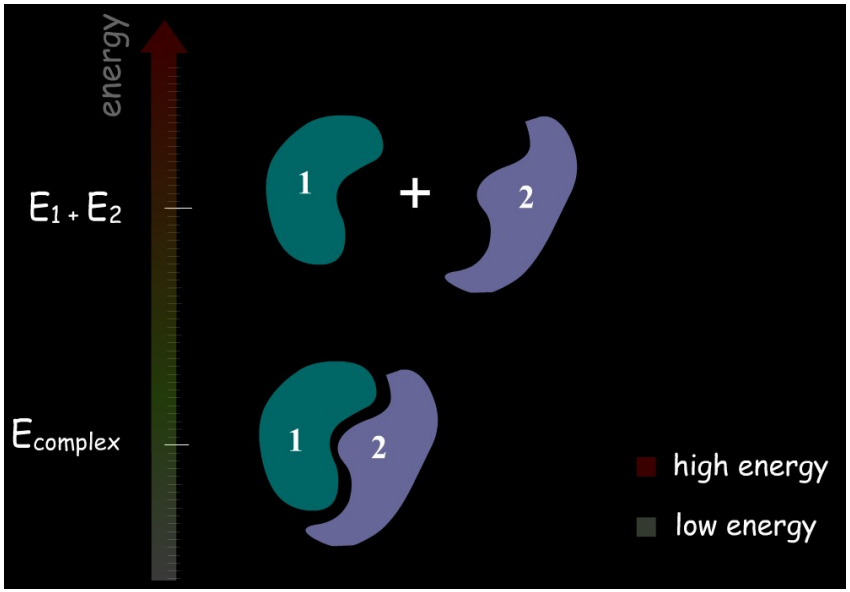
- Computational docking exploit the concept of molecular complementarity. The structures interact like a hand in a glove, where both the shape and the physico-chemical properties of the structures contribute to the fit.
- Shape complementarity is the primary criterion for evaluating the fit in the computational docking of two candidate structures
- In addition to shape compatibility, chemical and physico-chemical complementarity are also important criteria in the docking between candidate structures

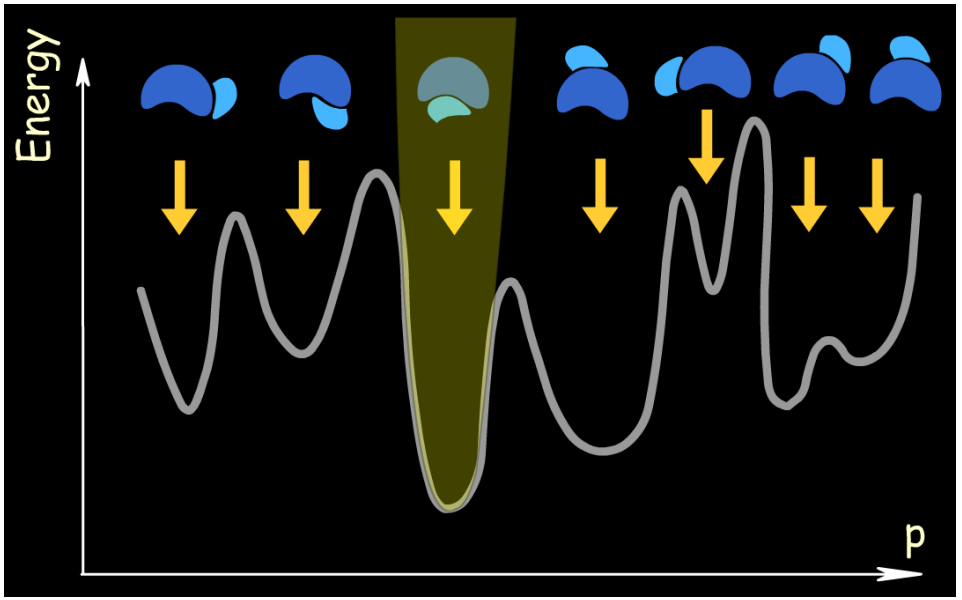




Energy Dictates Molecular Associations

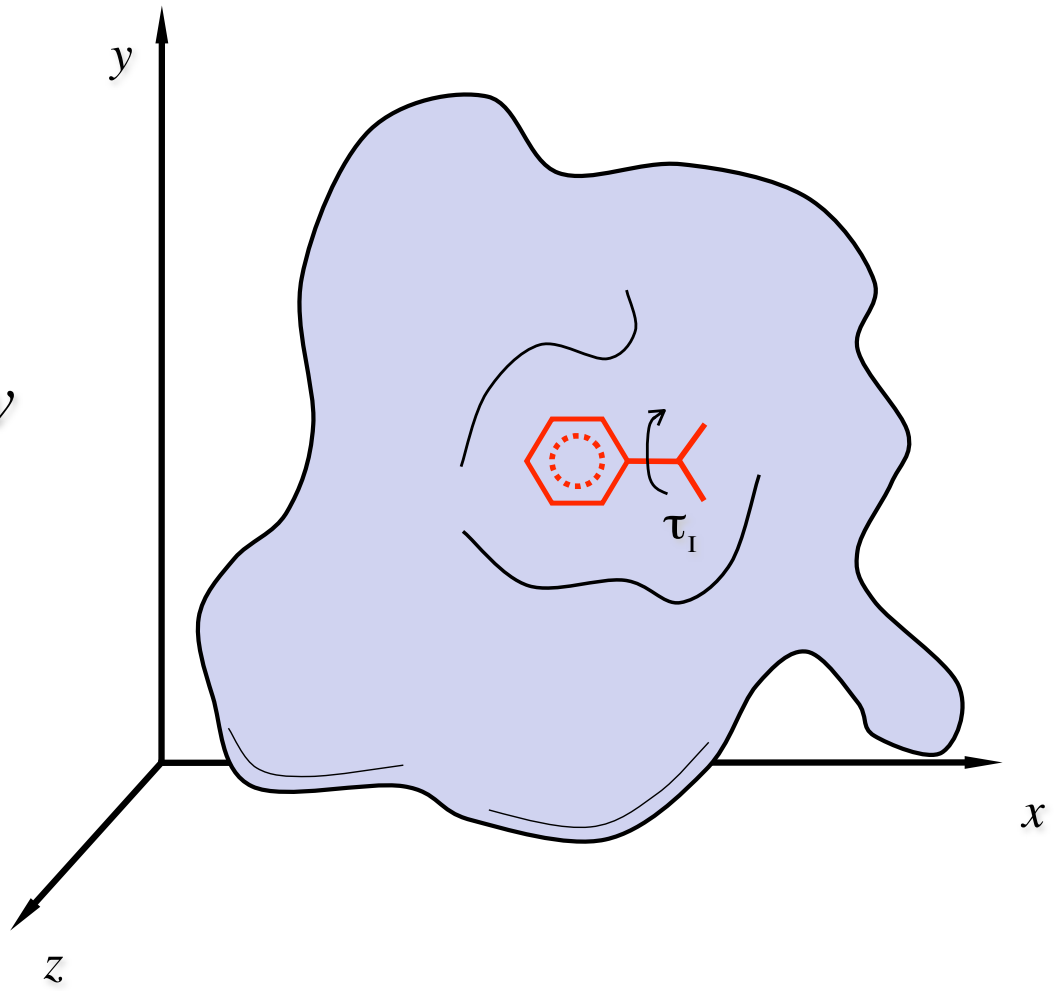
- The process of "self-assembly" is dictated by forces that are energy based: the complex has a lower potential energy than its constituent parts, and this keeps the parts together
- The goal of computational docking is to find the 3D configuration of the complex that minimizes the energy





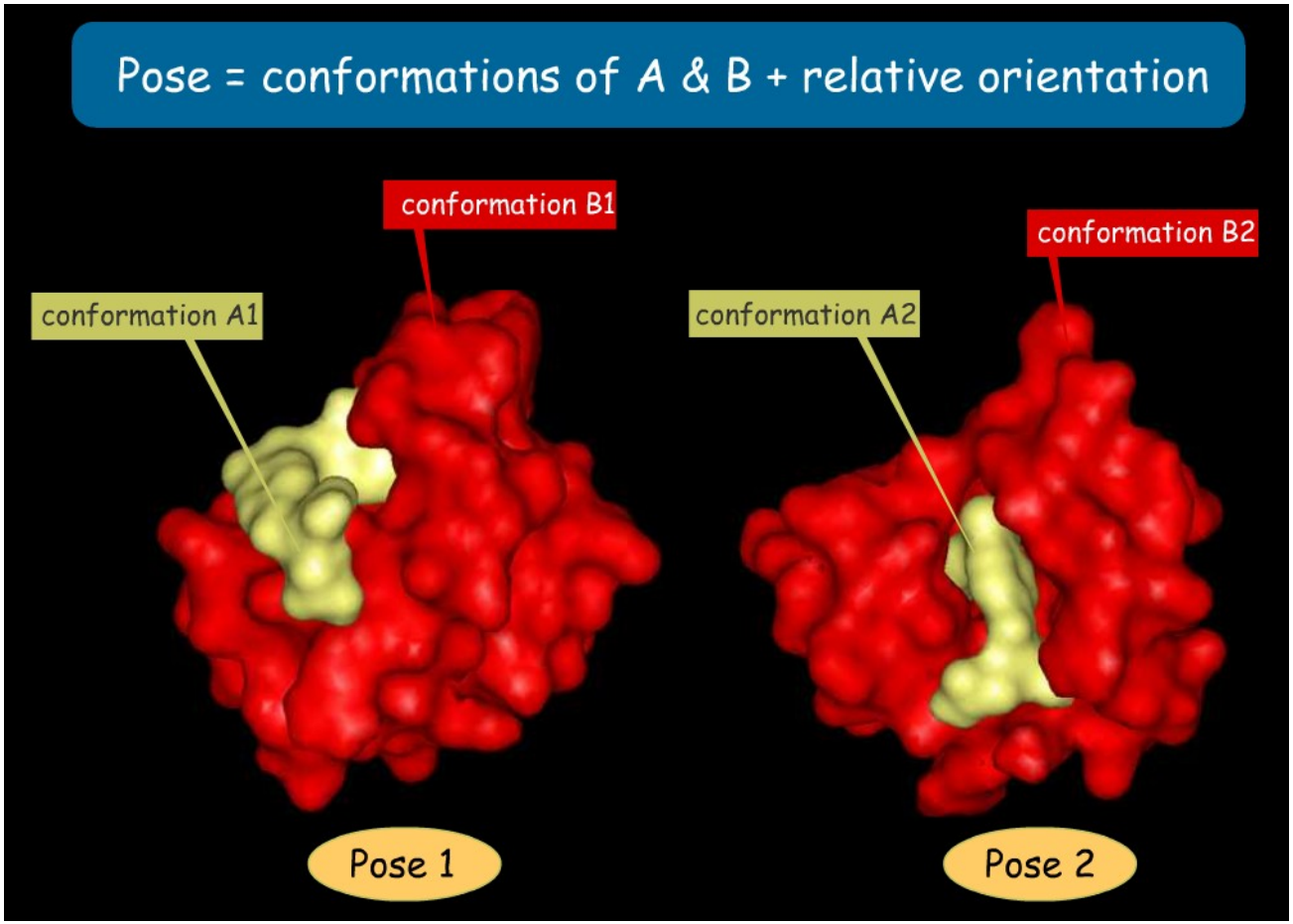
Defining a Docking

- Position
  - *x*, *y*, *z*
- Orientation
  - *qx*, *qy*, *qz*, *qw*
- Torsions
  - $\tau_1, \tau_2, \ldots \tau_n$



Definition of the "Pose"

- A "pose" is a term widely adopted for describing the geometry of a particular complex (also called "binding mode")
- It refers to a precise configuration which is characterized not only by the relative orientation of the docked molecules but also their respective conformations



Key aspects of docking...

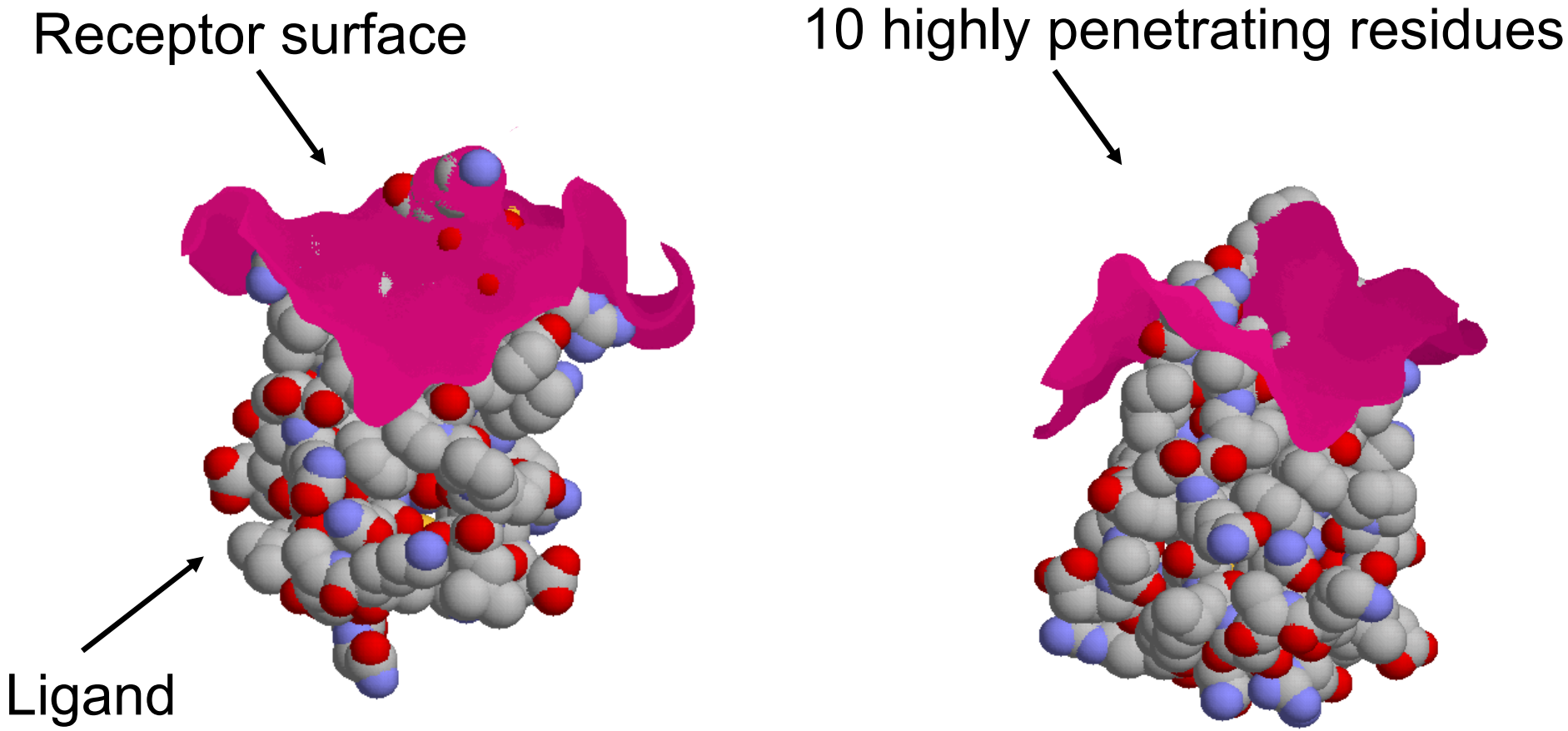
- Scoring Functions
  - *Predicting the energy of a particular pose*
  - *Often a trade-off between speed and accuracy*
- Search Methods
  - *Finding an optimal pose*
  - *Which search method should I use?*
- Dimensionality
  - *Can we trust the answer?*

Bound vs. Unbound Docking

- The simplest problem is the "bound" docking problem. The goal in this case is to reproduce a known complex where the starting point is atomic structures from a co-crystal

- The "unbound" docking problem is significantly more difficult task. Here the starting point is structure in their unbound conformation, perhaps the native structure, perhaps a modeled structure, etc.

Bound vs. Unbound

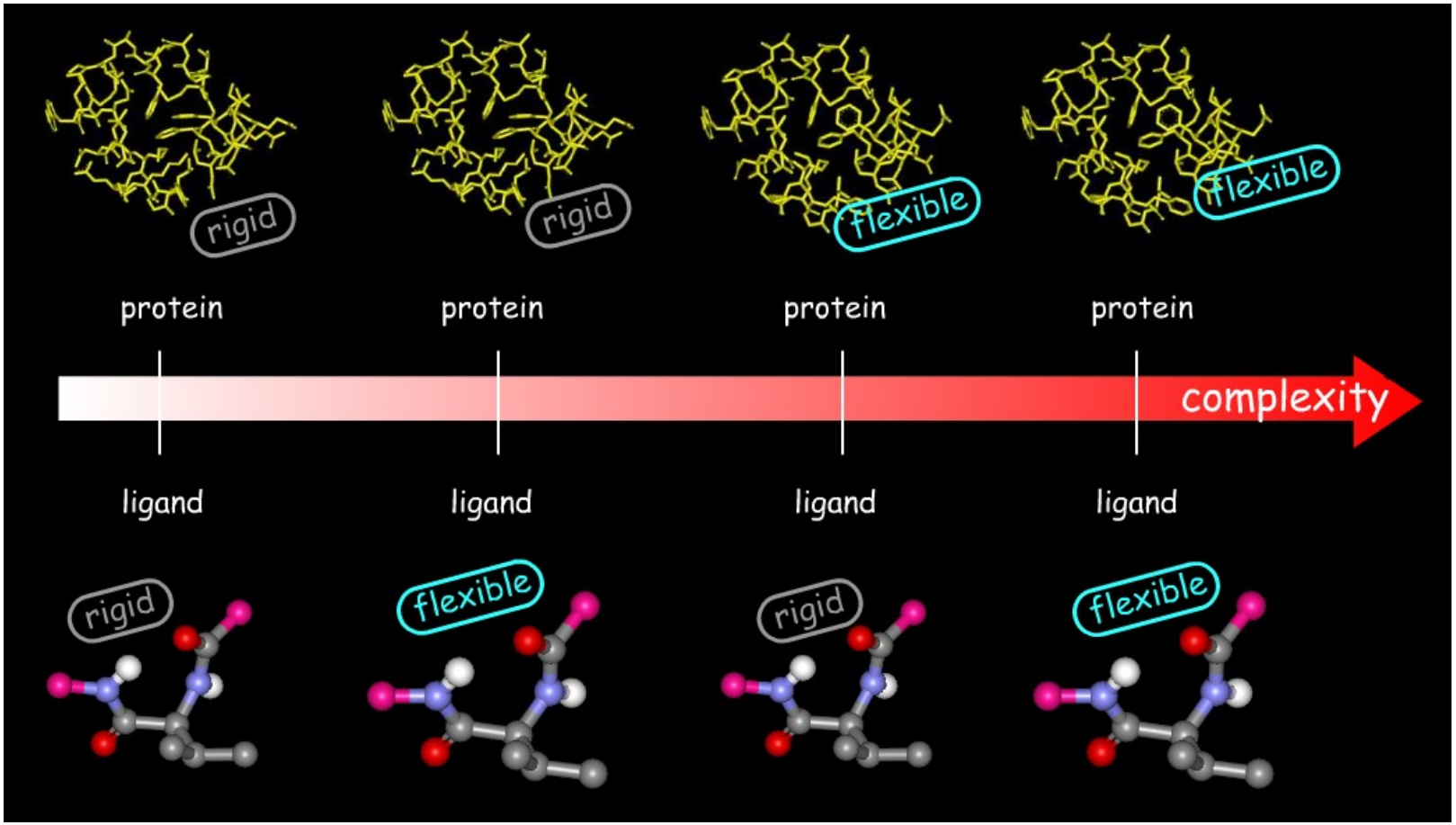


Kallikrein A/trypsin inhibitor
complex (PDB codes 2KAI,6PTI)

Molecular Flexibility in Protein-Ligand Docking

- The mutual adaptation of a ligand with its receptor is crucial to understanding ligand binding and protein function
- One of the major challenges in molecular docking is how to account for this adaptation in docking calculations
- The docking problem can be classified according to the way flexibility is modeled. In ascending order of complexity:
  1. Rigid body docking ignores the flexibility of the molecules and treats them like rigid objects
  2. Rigid receptor – flexible ligand docking: only the ligand is treated as flexible, receptor is rigid
  3. Flexible receptor – flexible ligand docking: both protein and ligand are treated as flexible.

Molecular Flexibility in Protein-Ligand Docking



Approach to the Problem

- One of the first suggestions on how to tackle this problem was given by Crick who postulated that “knobs” could be fitted into “holes” on the protein surface

- Lee & Richards had already described how to treat the van der Waals surface of a protein, however the van der Waals surface was not appropriate for docking purposes (too many crevices)

- Connolly showed how to calculate the molecular surface (and also the solvent accessible surface) which allowed for many docking programs to start being developed in the mid 1980’s

Docking Methodology

- All small molecule docking programs have three main components
  - Representation of system
  - The search algorithm
  - The scoring or energy evaluation routine
- In order to do a good job of docking, you need to search efficiently and evaluate the energy accurately

A Matter of Size

- For two proteins, the docking problem is very difficult since the search space (all possible relative conformations) is extremely large
- In the case of a small molecule (drug, peptide or ligand) binding to a protein, we have a chance of exploring the conformational space, at least for the small molecule
- Now we want to consider the case where we have limited or no *a priori* knowledge and the mode of binding

Protein-Protein Docking Methods

- The approaches to protein-protein docking have a lot in common with small molecule docking
- The methods are still based on the combination of search algorithm coupled to a scoring function
- The scoring functions are essentially the same (since we are still dealing with atomic interactions), however the major problem is that the conformational space we now need to search is massive

FFT Methods

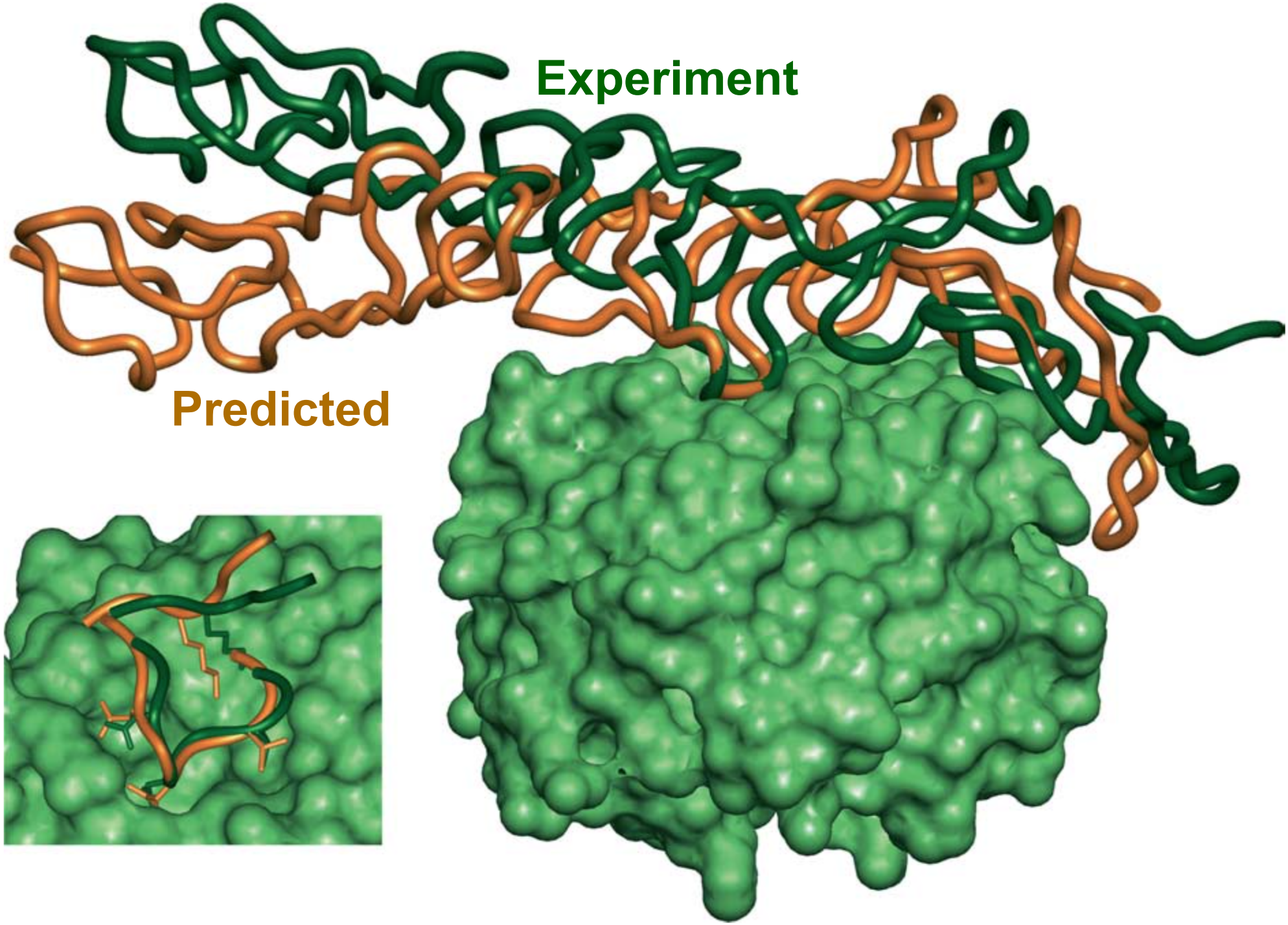
- A clever method developed by Katchalski-Katzir et al. uses FFT methods to search the translational and rotational degrees of freedom

- H. A. Gabb, R. M Jackson and M. Sternberg "Modelling protein docking using shape complementarity, electrostatics and biochemical information"
  J. Mol Biol. (1997) 272:106-120

Katchalski-Katzir Method

- Exhaustive enumeration of the rotational and translational degrees of freedom
- Since the scoring function is represented as a convolution, the enumeration of the translational degrees of freedom is efficient from exploiting FFT methods, reduces $O(N^6)$ to $O(N^3 \log N^3)$
- Implemented in the MolFit and FTDock programs, geometric, but extensions include electrostatics
- Time consuming, yet conceptually simple and appealing; often used for initial screening
- Volumetric – does not use a MS representation

Nidogen-Laminin Complex



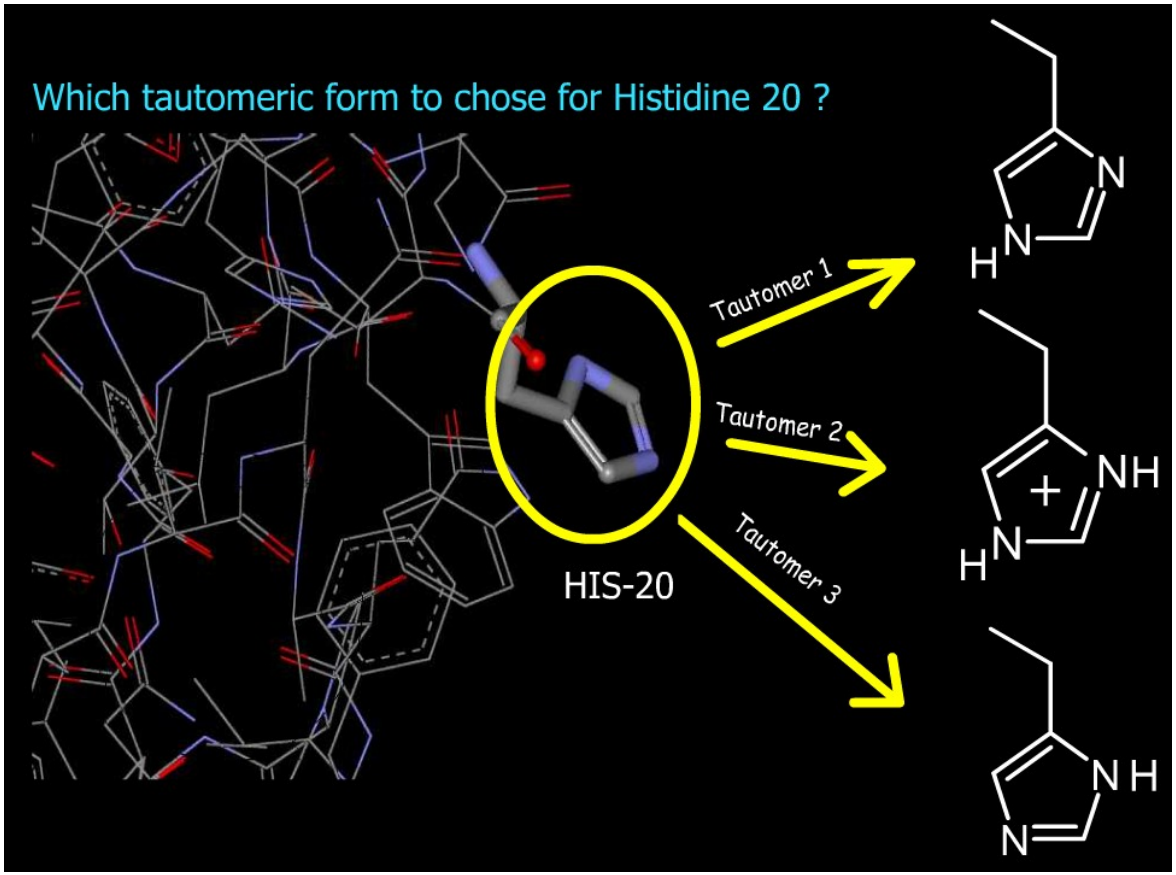
Representation of the System
or
Site Characterization

Representation of Phase Space

- The type or choice of representation for a system is really a reflection of the type of energy evaluation or scoring function that will be used
- If we would choose the most straightforward or logical representation of the atomic coordinates of the of the two biomolecules, we would simply use a molecular mechanics force field such as Amber, CHARMM or OPLS
- However, this may not be the best choice in terms of computational efficiency or practicality

Protein Preparation

- The preparation of the protein calls for great care. Important decisions include the choice of the tautomeric forms of histidine residues, the protonation states of amino-acids and conformations of some residues; their incorrect assignments may lead to docking errors.

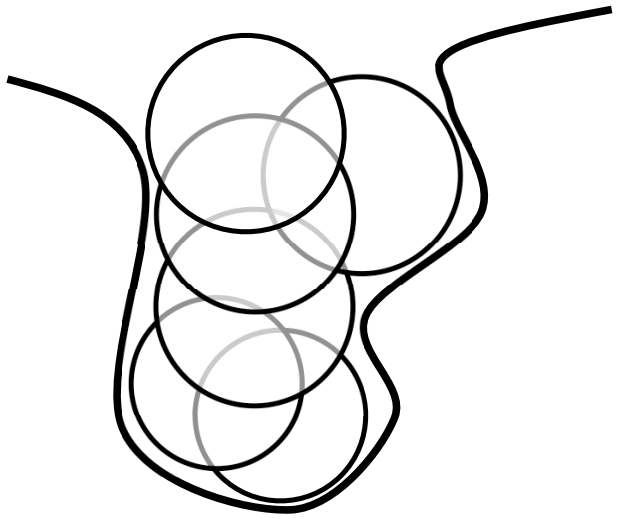


Small Molecule Preparation

- Before generating and docking the 3D structures of a library of ligands, it is important to "clean up" the 2D structures being used by removing any counter ions, salts, or water molecules that might be part of the registered structure
- All reactive or otherwise undesirable compounds must also be removed
- Possibly generate all optical isomers (enantiomers), cis/trans isomers, tautomers, and protonation states of the structures
- For most docking programs the tautomeric and protonation state of the ligands to be docked is defined by the user; in general the structure considered to be dominant at a neutral pH is generated; here also, incorrect assignments may lead to docking errors

DOCK

- The DOCK program is from the Kuntz group at UCSF
- It was the first docking program developed in 1982
- It represents the (negative image of the) binding site as a collection of overlapping spheres

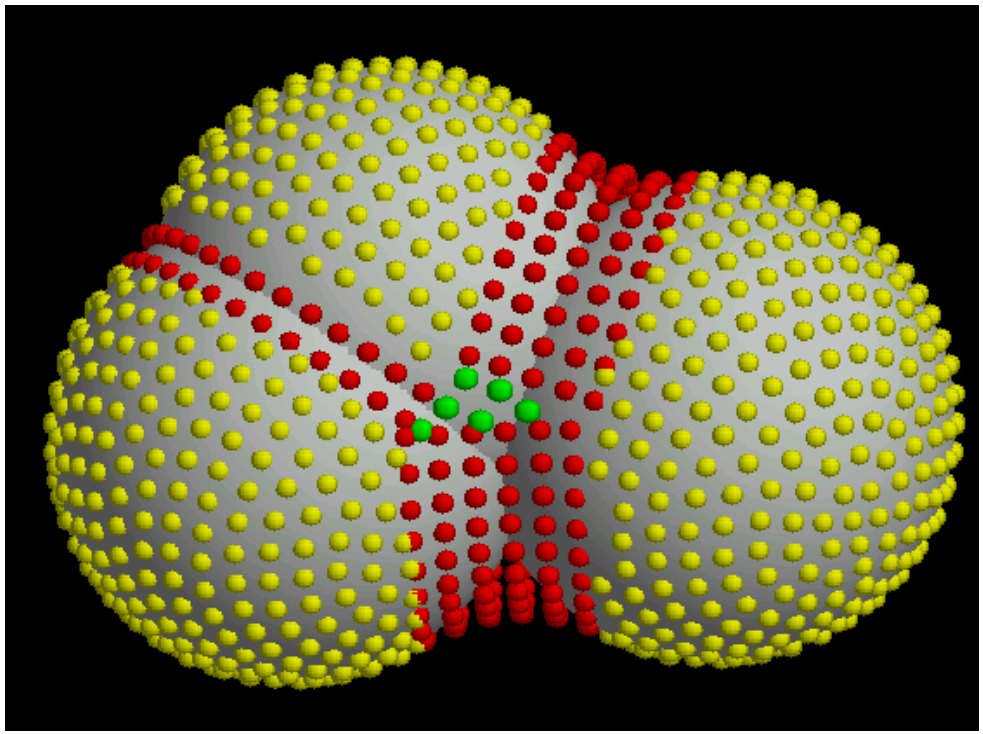


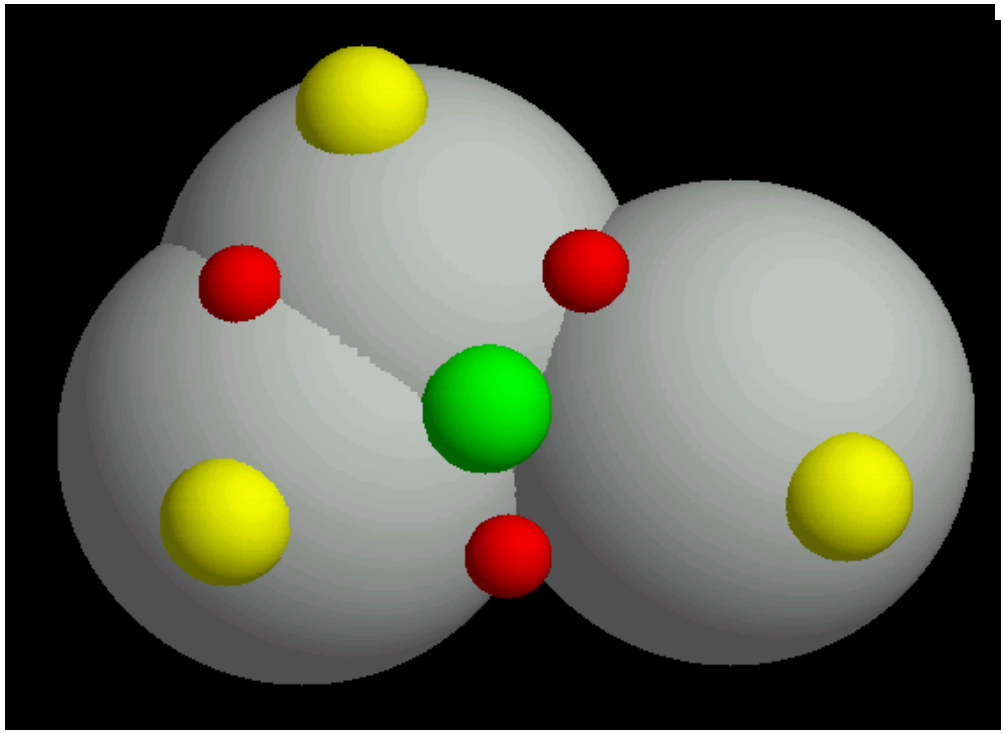
DOCK

- This method of a negative image is targeted at finding complexes with a high degree of shape complementarity
- The ligand is fit into the image by a least squares fitting of the atomic positions to the sphere centers
- Creating the negative image is obviously not a problem with a unique solution. Hence, factors such as the sphere radius and center-to-center distance of the spheres must be carefully controlled.

Surface Representation

- Dense MS surface (Connolly)
- Sparse surface (Shuo Lin et al.)



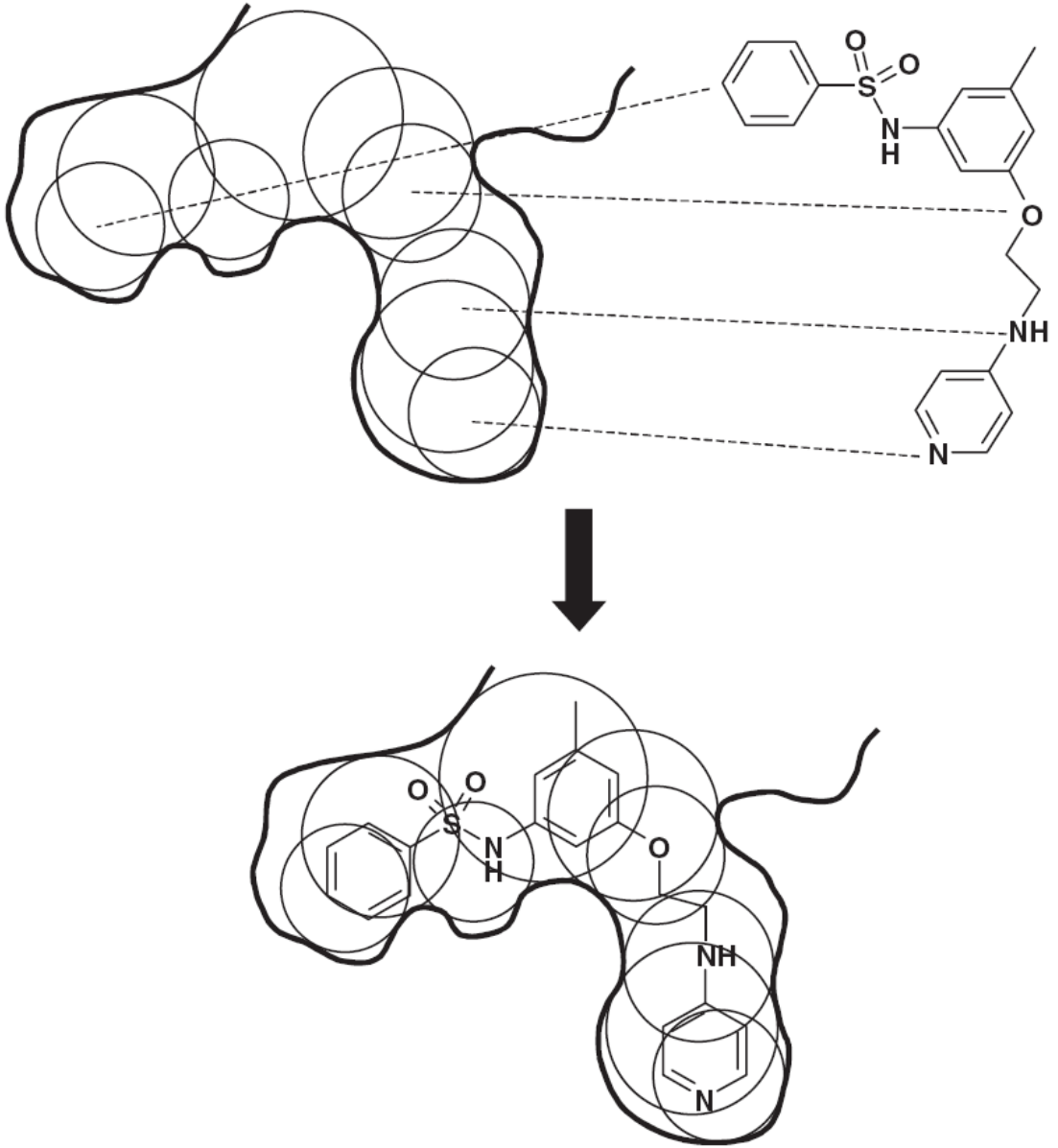


The DOCK algorithm – Rigid docking

- The DOCK algorithm developed by Kuntz and co-workers is generally considered one of the major advances in protein–ligand docking [Kuntz et al., *JMB,* **1982**, *161*, 269]

- The earliest version of the DOCK algorithm only considered rigid body docking and was designed to identify molecules with a high degree of shape complementarity to the protein binding site.

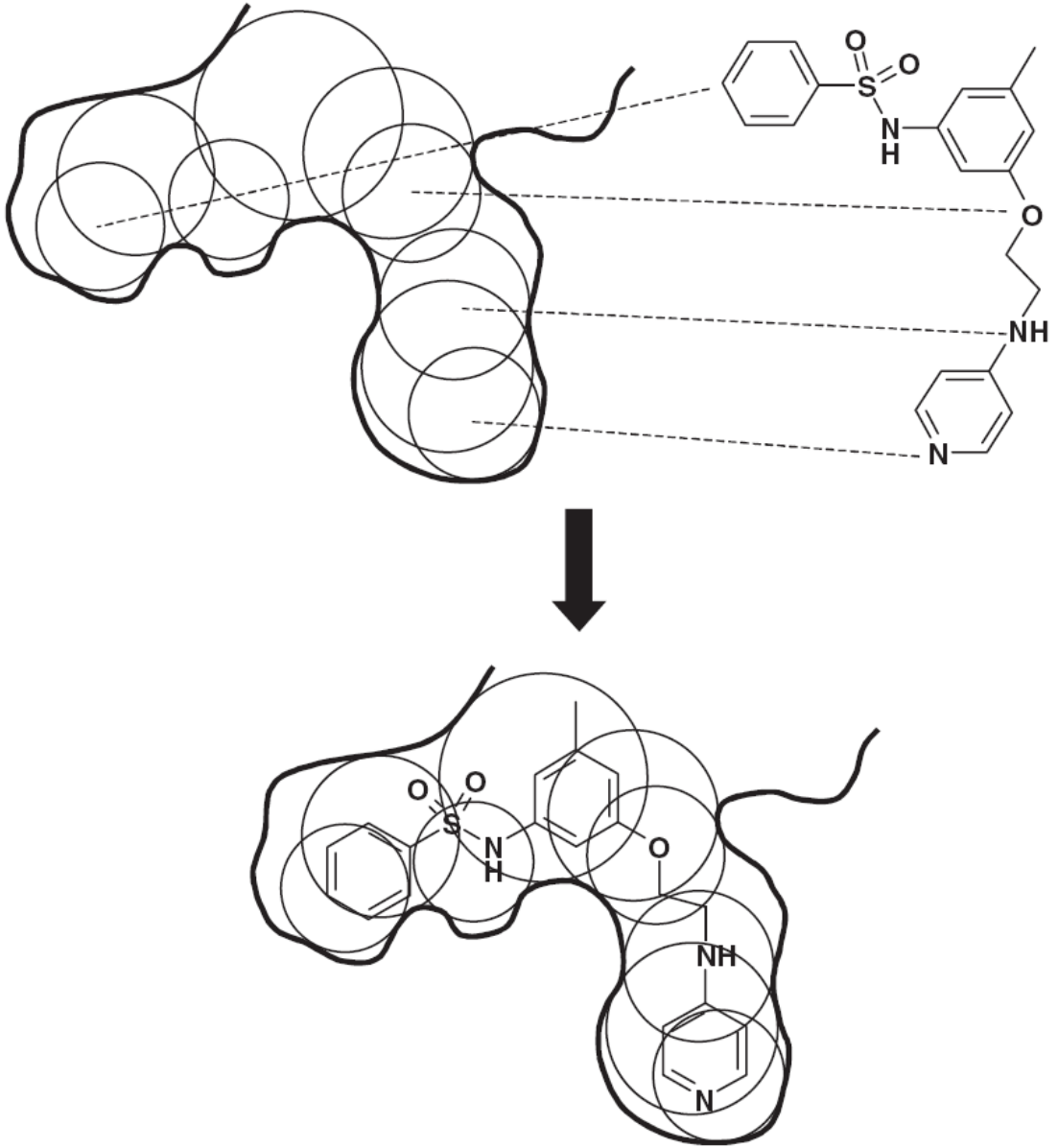
- The first stage of the DOCK method involves the construction of a “negative image” of the binding site consisting of a series of overlapping spheres of varying radii, derived from the molecular surface of the protein



AR Leach, VJ Gillet, An Introduction to Cheminformatics

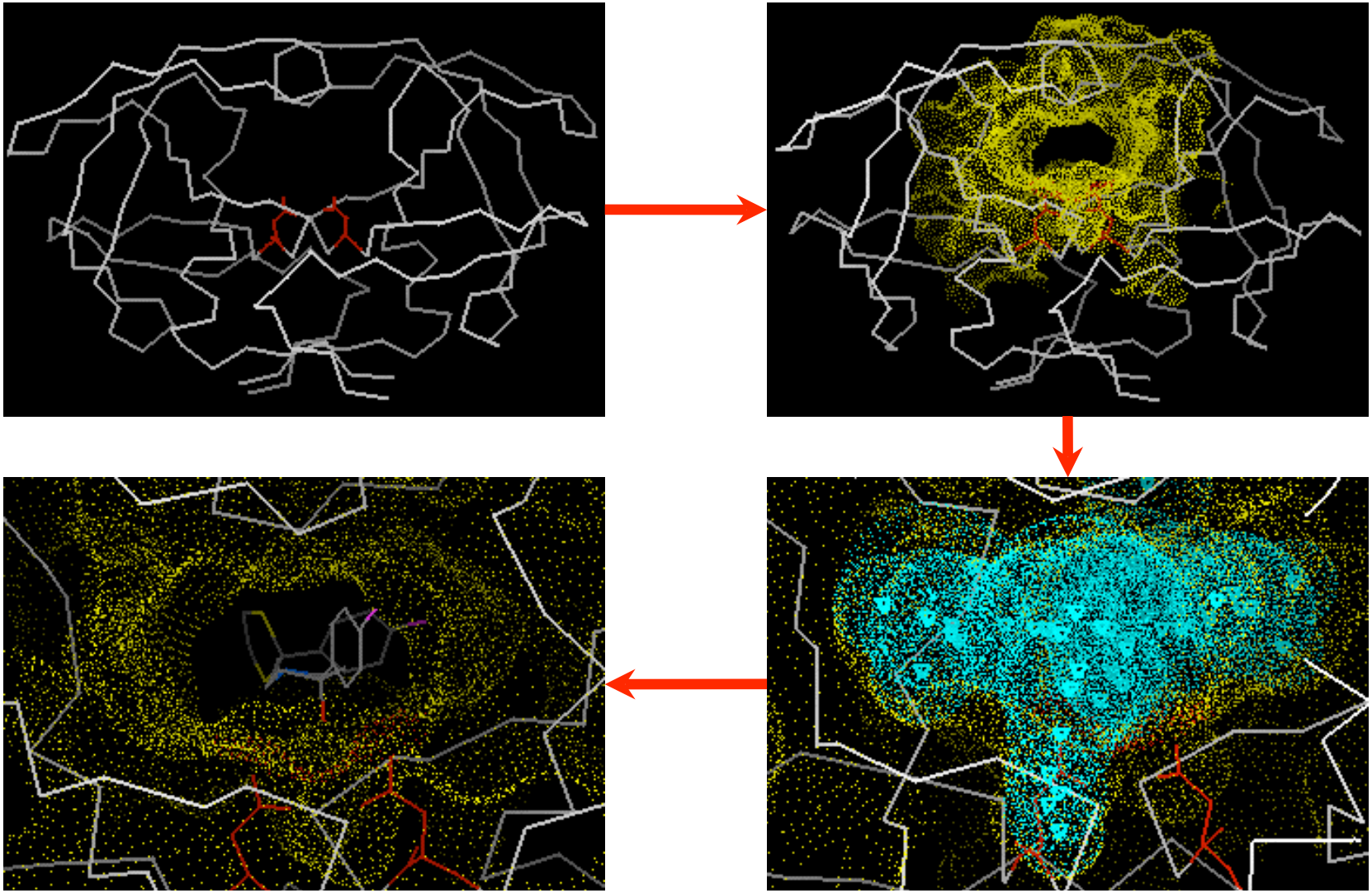
The DOCK algorithm – Rigid docking

- Ligand atoms are then matched to the sphere centres so that the distances between the atoms equal the distances between the corresponding sphere centres, within some tolerance.
- The ligand conformation is then oriented into the binding site. After checking to ensure that there are no unacceptable steric interactions, it is then scored.
- New orientations are produced by generating new sets of matching ligand atoms and sphere centres. The procedure continues until all possible matches have been considered.



AR Leach, VJ Gillet, An Introduction to Cheminformatics

DOCK



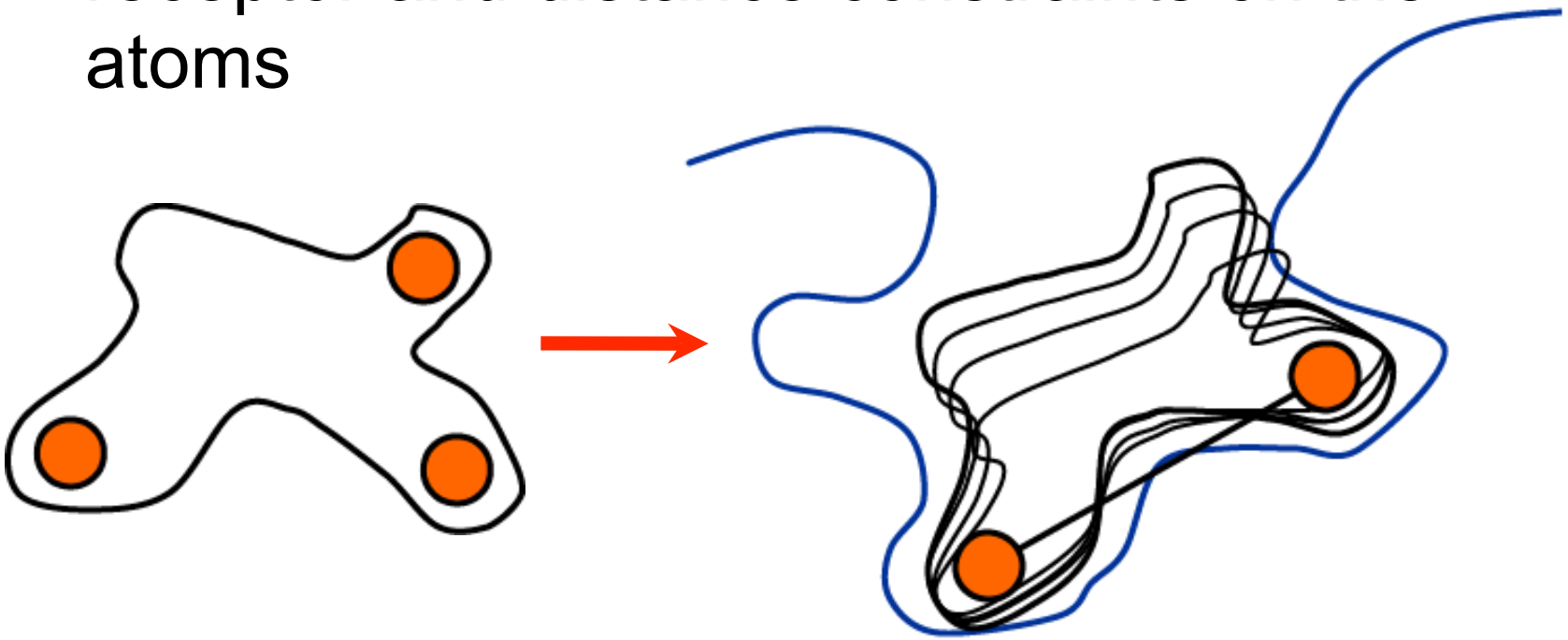
http://dock.compbio.ucsf.edu

DOCK Extensions

- Dock 4.0 has recently been extended to include:
  - Several different scoring schemes
  - Ligand flexibility (via incremental construction and fragment joining)
  - Chemical properties of receptor (each sphere assume a chemical characteristic)

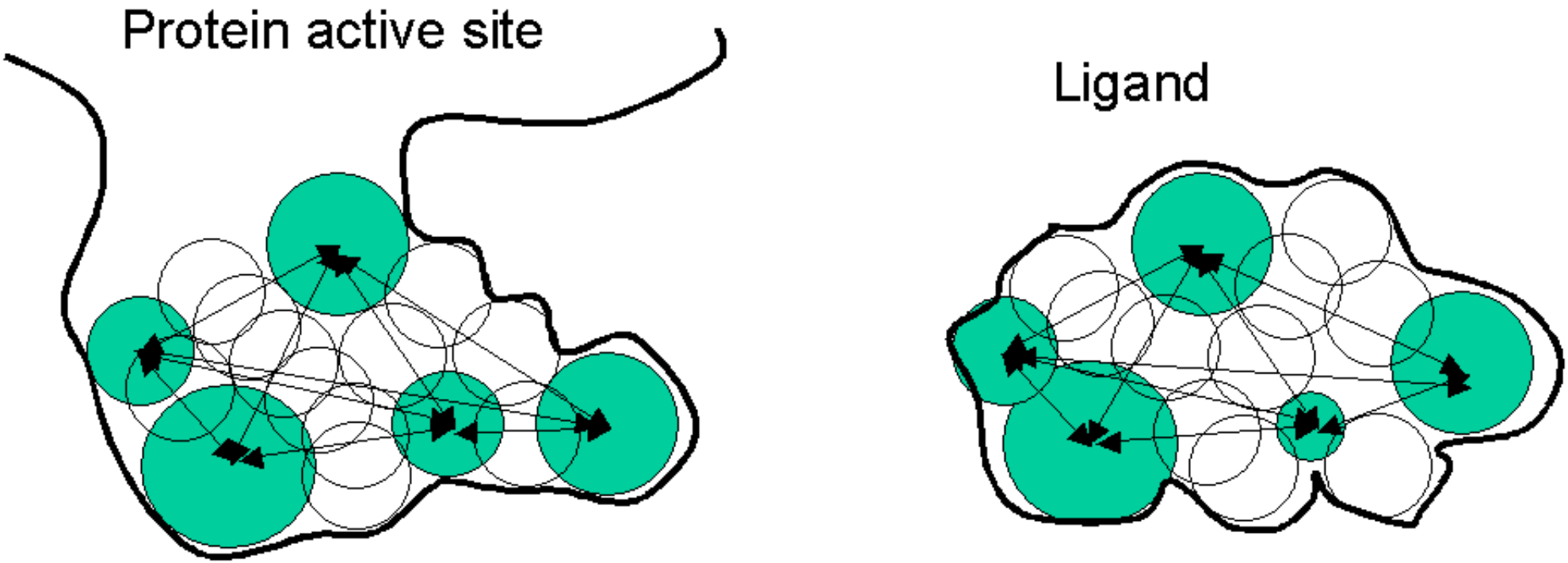
CLIX

- CLIX uses a chemical description of the receptor and distance constraints on the atoms



Lawrence et al., *Proteins: Struct. Func. And Gen.* **12**:31 (1992)

Clique-Detection



- Nodes comprise of matches between protein and ligand
- Edges connect distance compatible pairs of nodes
- In a clique all pair of nodes are connected

ESCHER

- ESCHER uses the solvent accessible surface from a Connolly algorithm
- This surface is cut into 1.5 Å slabs that are transformed into polygons and used for rigid docking (again image matching)

G. Ausiello, G. Cesareni, M. Helmer Citterich, "ESCHER: a new docking procedure applied to the reconstruction of protein tertiary structure", *Proteins*, **28**:556-567 (1997)

Search Methods

Search Basics

- One of the more difficult tasks in computational docking is simply enumerating the number of ways two molecules can be put together (3 translational plus 3 rotational degrees of freedom)
- The size of this search space grows exponentially as we increase the size of the molecules, however this is still only for rigid structures

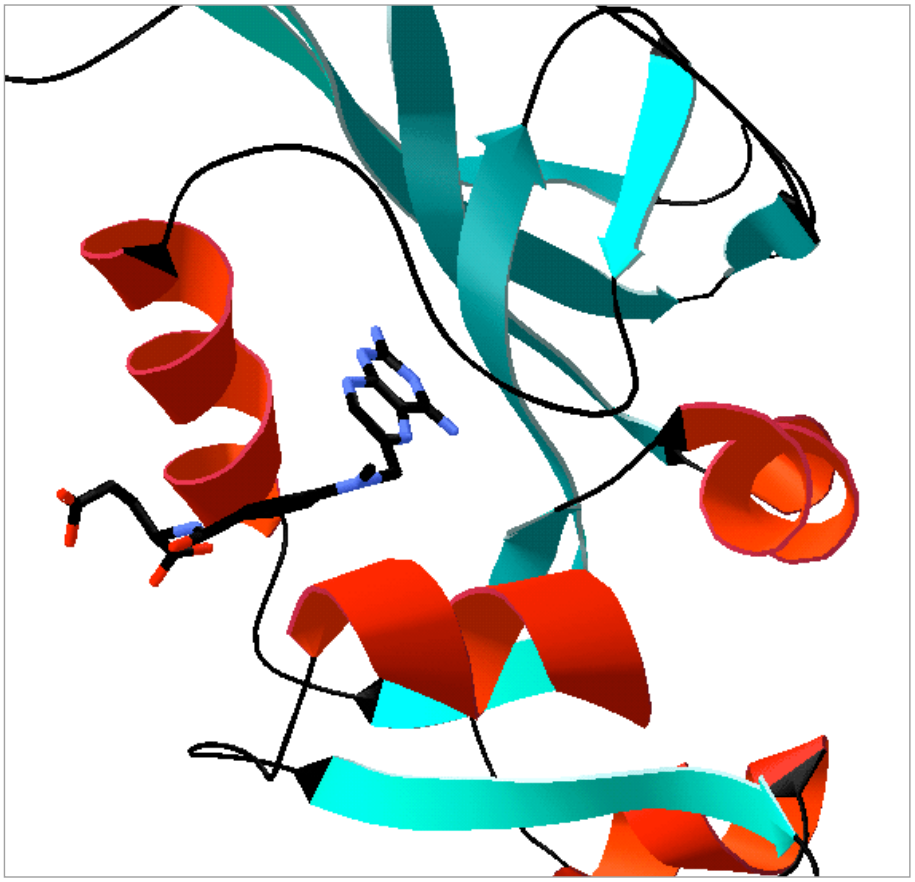
Sampling Hyperspace

- Say we are searching in D-dimensional hyperspace...
- We want to evaluate each of the D dimensions N times.
- The number of "evals" needed, n, is: $n = N^D$

  $\therefore N = n^{1/D}$
- For example, if $n = 10^6$ and...
  - $D=6$, $N = (10^6)^{1/6} = 10$ evaluations per dimension
  - $D=36$, $N = (10^6)^{1/36} = \sim 1.5$ evaluations per dimension
- Clearly, the more dimensions, the tougher it gets.

SAMPLING HYPERSPACE !

Very CPU time consuming...



Dihydrofolate reductase with a methotrexate (4dfr.pdb)

$$N=T^{360/i}$$

N: number of conformations

T: number of rotable bonds

I: incremental degrees

Methotrexate

10 rotable bonds

30° increments (discrete)

10^{12} plausible conformations!

Search Difficulties

- If we allow flexibility of one or both of the binding partners, this quickly becomes an intractable problem
- If we manage to solve/circumvent the problem of flexibility, we then want to be able to screen large databases of structures or drugs, so our troubles are again compounded

Spectrum of Search: Breadth and Level-of-Detail

Search Breadth

- Local
  - Molecular Mechanics (MM)
- Intermediate
  - Monte Carlo Simulated Annealing (MC SA)
  - Brownian Dynamics
  - Molecular Dynamics (MD)
- Global
  - Docking

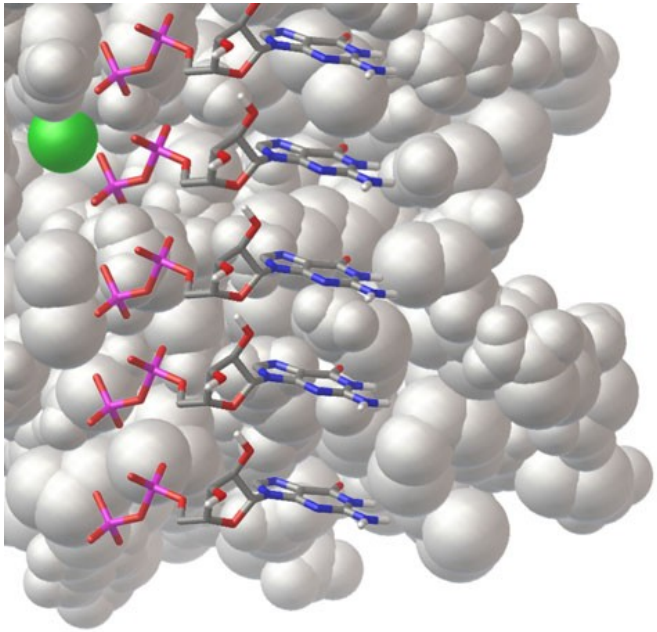
Level-of-Detail

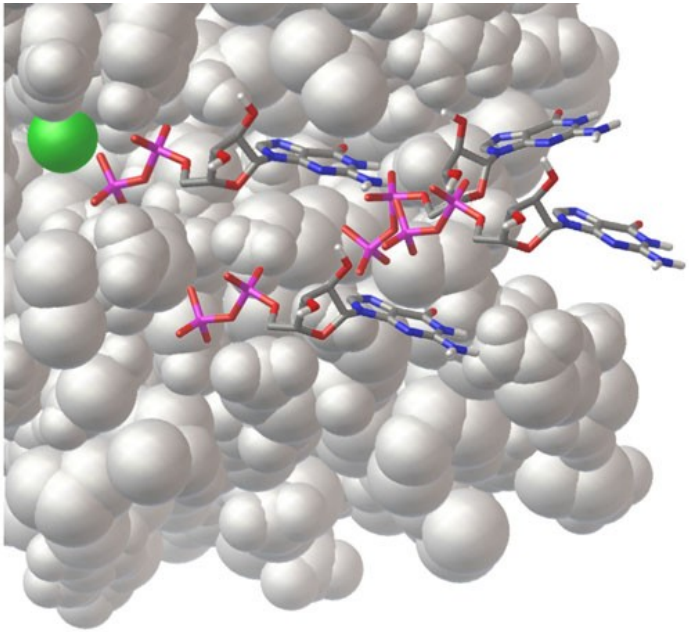
- Atom types
- Bond stretching
- Bond-angle bending
- Rotational barrier potentials

- Implicit solvation
- Polarizability

- What's rigid and what's flexible?

Two Kinds of Search





Systematic

- Exhaustive, deterministic
- Outcome is dependent on granularity of sampling
- Feasible only for low-dimensional problems

Stochastic

- Random, outcome varies
- Must repeat the search or perform more steps to improve chances of success
- Feasible for larger problems

Stochastic Search Methods

- Simulated Annealing (SA)*
- Evolutionary Algorithms (EA)
  - Genetic Algorithm (GA)*
- Others
  - Tabu Search (TS)
  - Particle Swarm Optimisation (PSO)
- Hybrid Global-Local Search Methods
  - Lamarckian GA (LGA)*

*Supported in AutoDock

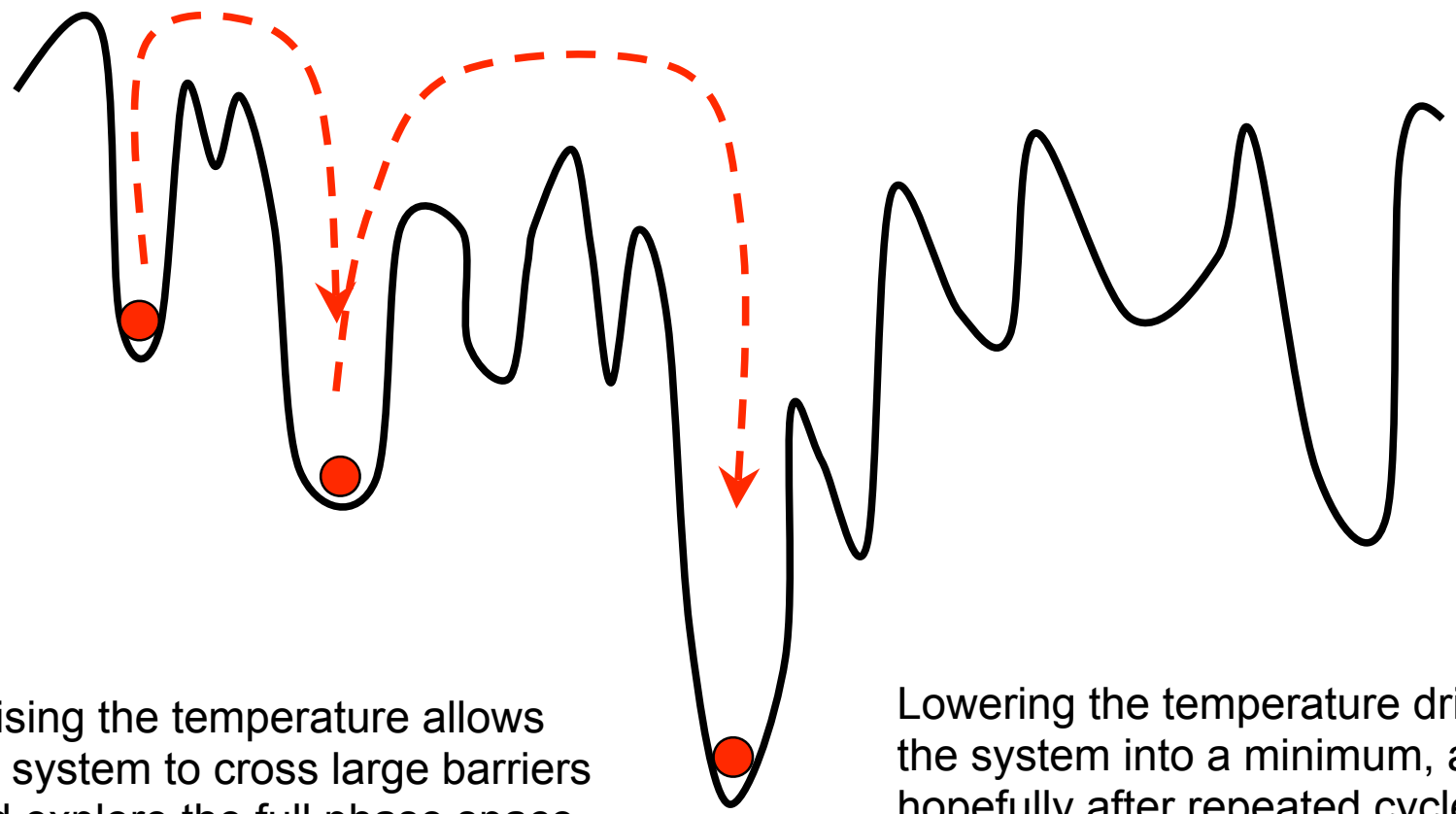
Monte Carlo Simulated Annealing

- Also known as Metropolis Monte Carlo
- The basic steps are
  1. The ligand performs a random walk around the protein
  2. At each step, a random displacement, rotation, etc. is applied and the energy is evaluated and compared to the previous energy
  3. If the new energy is lower, the step is accepted
  4. If the energy is higher, the step is accepted with a probability $\exp(\Delta E/kT)$

Simulated Annealing

- If we were to perform this at a constant temperature, it would be basic Monte Carlo
- In this case, after a specified number of steps, the temperature is lowered and the search repeated
- As the temperature continues to go down, steps which increase the energy become less likely and the system moves to the minimum (or minima)

Simulated Annealing

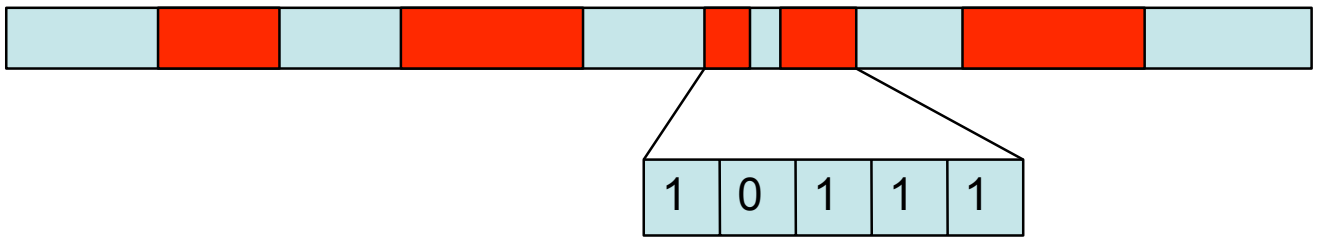


Raising the temperature allows the system to cross large barriers and explore the full phase space

Lowering the temperature drives the system into a minimum, and hopefully after repeated cycles, the global minimum

Genetic Algorithms

- Genetic algorithms belong to a class of stochastic search methods, but rather than operating on a single solution, they function on a population
- As implied by the name, you must encode your solution, structure or problem as a genome or chromosome
- The translation, orientation and conformation of the ligand is encoded (the *state variables*)



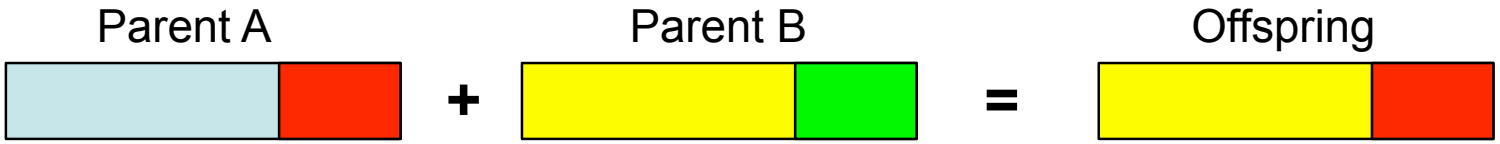
Genetic Algorithms

- The algorithm starts by generating a population of genomes and then applies crossover and mutation to the individuals to create a new population
- The “fitness” of each structure has to be evaluated, in our case by our estimate of the binding free energy
- The best member or members survive to the next generation
- This procedure is repeated for some number of generations or energy evaluations

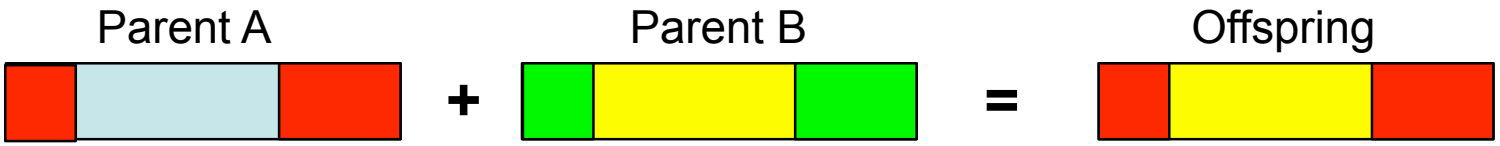
Crossover and Mutation

- There are several possible methods of crossover

Single Point Crossover

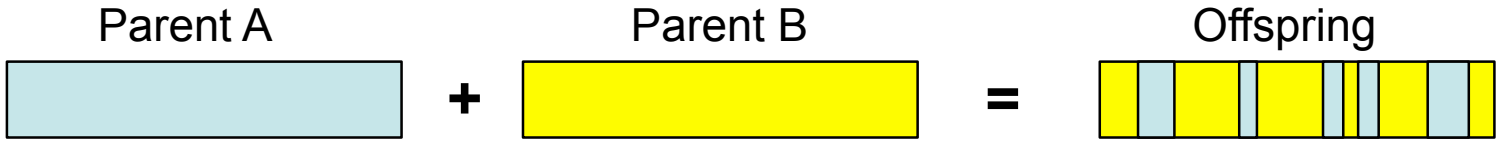


Two Point Crossover



Crossover and Mutation

Uniform Crossover

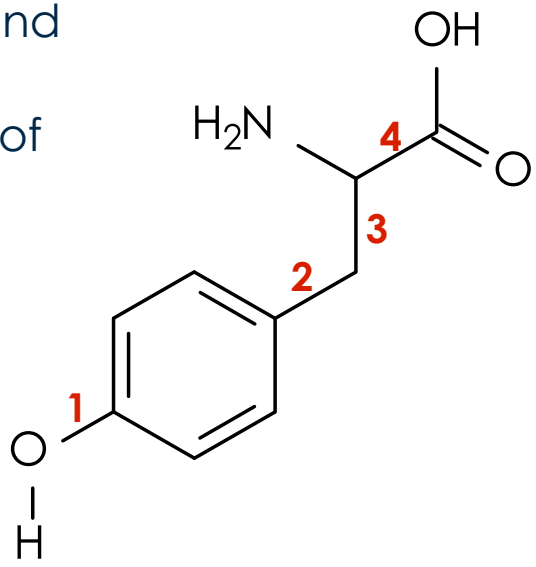


- For mutation, selected bits are simply inverted

Structure Representation in a Genetic Algorithm

Use of a Genetic Algorithm as a sampling method

- Each conformation is described as a set of rotational angles.
- 64 possible angles are allowed to each of the bond in the ligand.
- Each plausible dihedral angle is codified in a set of binary bits (2^6=64)
- Each conformation is codified by a so called chromosome with 4 × 6 bits (0 or 1)



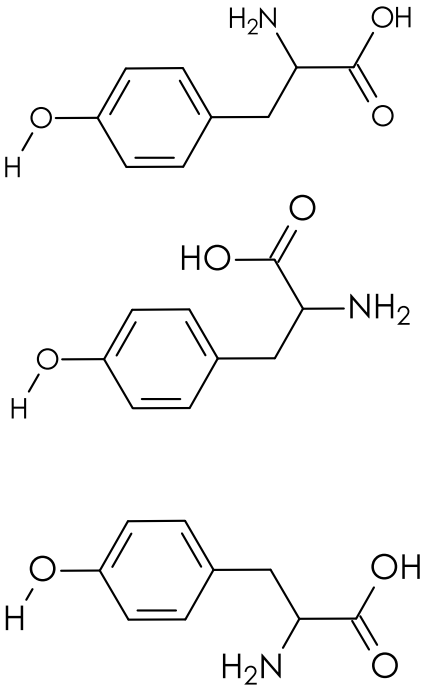
111010.010110.001011.010010

Φ_1 Φ_2 ...

$$\Phi_1 = 1\times2^5 + 1\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 0\times2^0 = 58°$$

Population of Structures

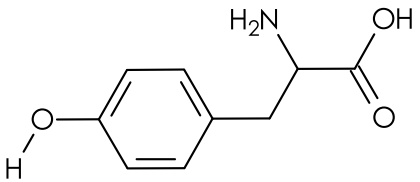
Population (ie, set of chromosomes or configurations)

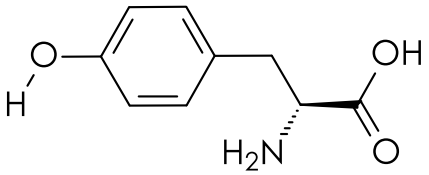


011010.010110.011010.010111 ← Chromosome
111010.010110.001011.010010
001010.010101.000101.010001
101001.101110.101010.001000
001010.101000.011101.001011

Gene

Discrete Mutation Operator



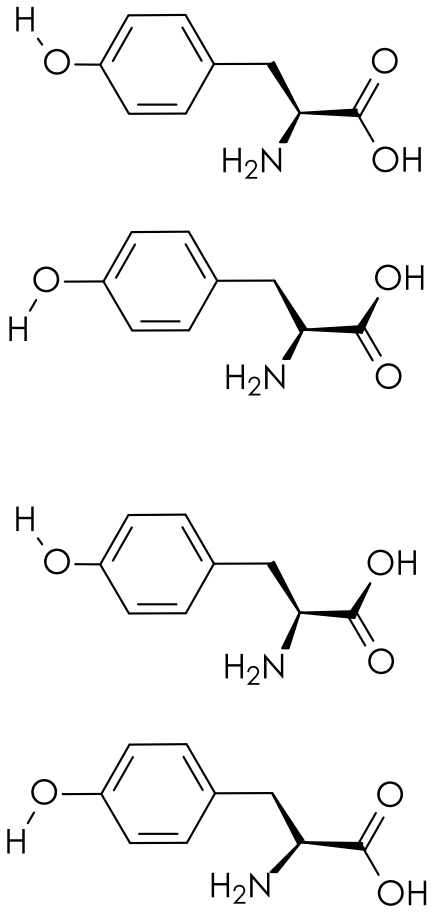


011010.010110.011010.010111

Single Mutation

011010.011110.011110.010111

Crossover Operator



001010.010101.000101.010001

011010.010110.011010.010111

Recombination
(Crossover)

001010.010101.011010.010111

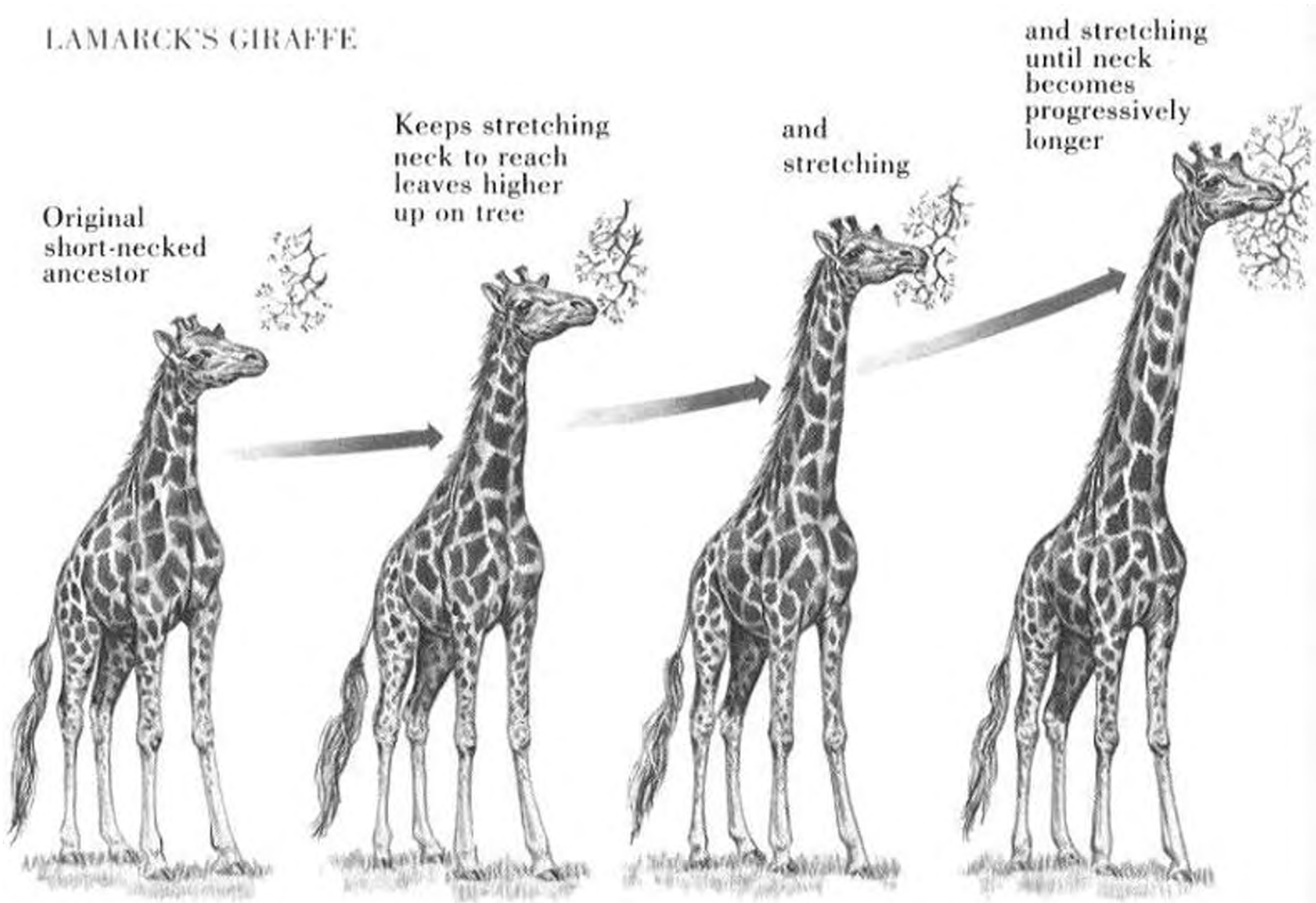
011010.010110. 000101.010001

Migration between Populations

Efficiency

- Random mutations are also possible (perhaps the orientation or position is shifted by some random amount)
- Binary crossover and mutation can introduce inefficiencies into the algorithm since they can easily drive the system away from the region of interest
- For this reason, genetic searches are often combined with local searches (producing a Lamarckian Genetic Algorithm)

LAMARCK'S GIRAFFE



Local Searches

- In a Lamarckian GA, the genetic representations of the ligands starting point is replaced with the results of the local search
- The mapping of the local search *back* on to the genetic representation is a biologically unrealistic task, but it works well and makes the algorithm more efficient
- One of the more common local search methods is the Solis and Wets algorithm

Solis and Wets Algorithm

Starting point x
Set bias vector b to 0
Initialize ρ

While max iterations not exceeded:

Add deviate d to each dimension
(from distribution with width ρ)

If new solution is better:

success++
b = 0.4 d + 1.2 b

else

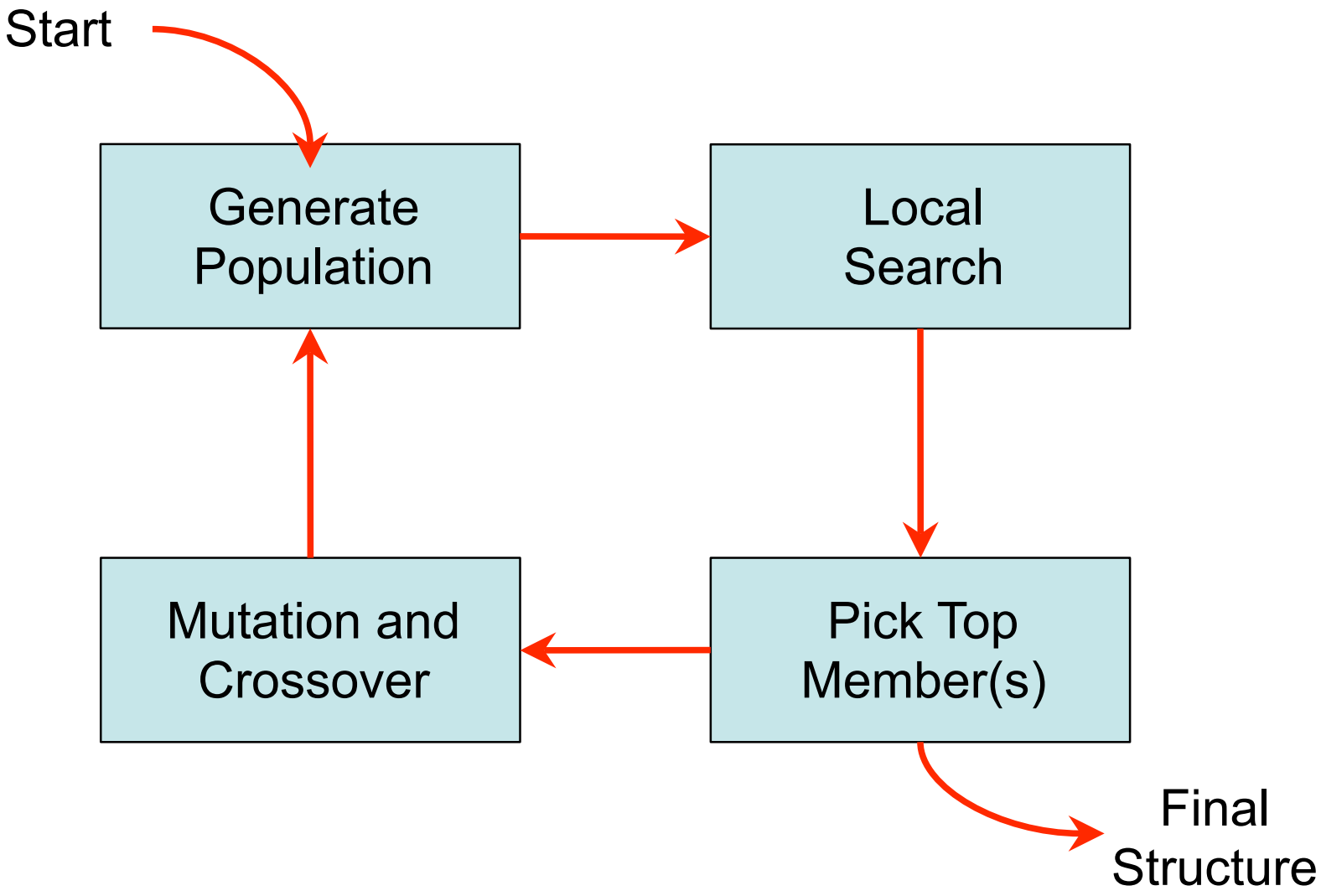
failures++
b = b - 0.4 d

Adaptive step size which biases the search in the direction of success

If too much success: increase ρ

If too many failures: decrease ρ

LGA Scheme

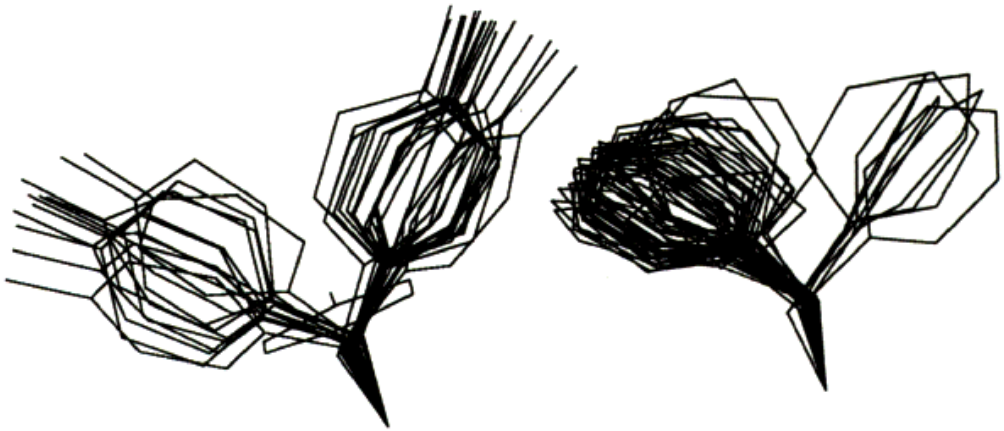

Start
Generate Population
Local Search
Pick Top Member(s)
Mutation and Crossover
Final Structure

Approaches to Flexibility

- A relatively simple molecule with 10 rotatable bonds has more than 10^9 possible conformation if we only consider 6 possible positions for each bond
- Monte Carlo, Simulated Annealing and Genetic Algorithm can help navigate this vast space
- Other methods have been developed to again circumvent this problem

Side Chain Rotamer Libraries

- Exploring the conformational space of protein side chains is a complex optimization problem that leads to a combinatorial explosion of conformers
- Protein side chains can be represented adequately by a small set of discrete rotamers
- Analysis of side-chain Ramachadran plots of pdb structures show that 17 of the 20 amino acids can be represented adequately by 67 side-chain rotamers
- This approach greatly simplified the problem and enabled side-chain flexiblity to be tackled

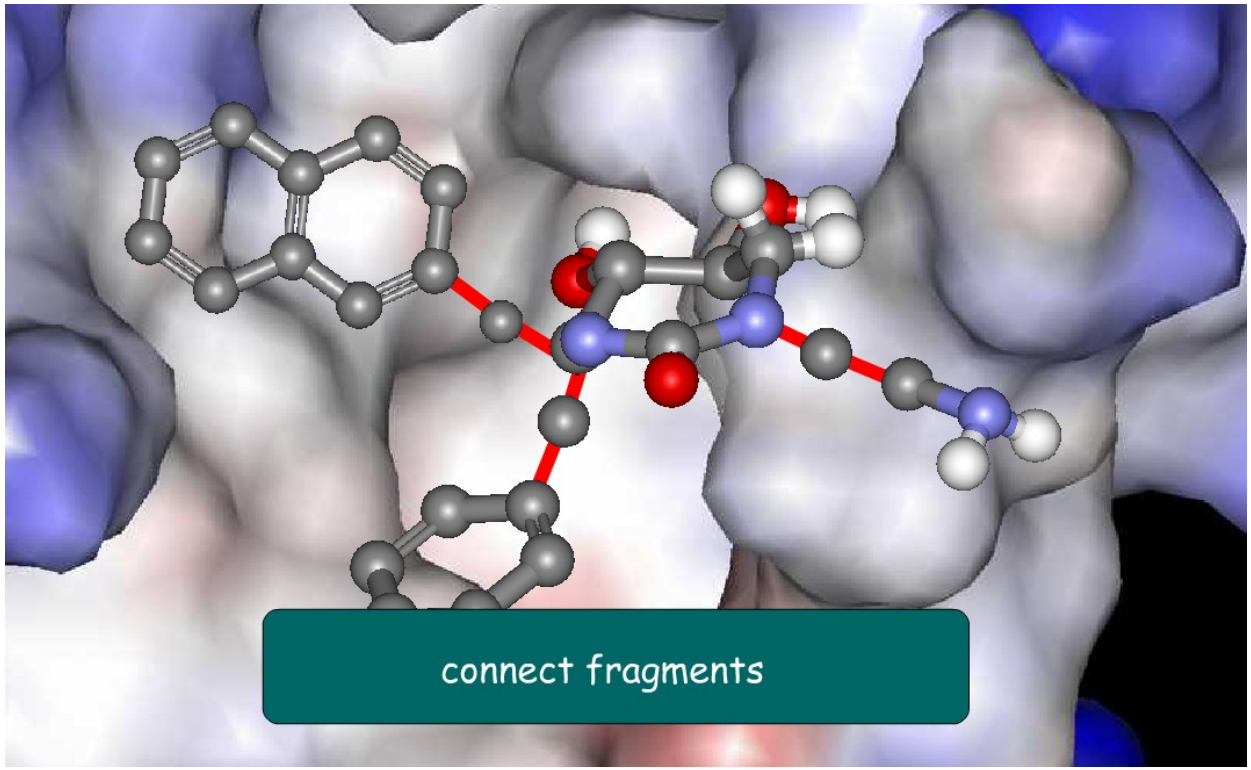


Flexibility

- Some algorithms (call Place & Join algorithms) break the ligand up into pieces, dock the individual pieces, and try and reconnect the bound conformations
- FlexX uses a library of precomputed, minimized geometries from the Cambridge database with up to 12 minima per bond. Sets of alternative fragments are selected by choosing single or multiple pieces in combination
- Flexible docking via molecular dynamics with minimization can handle arbitrary flexibility, however it is extremely slow

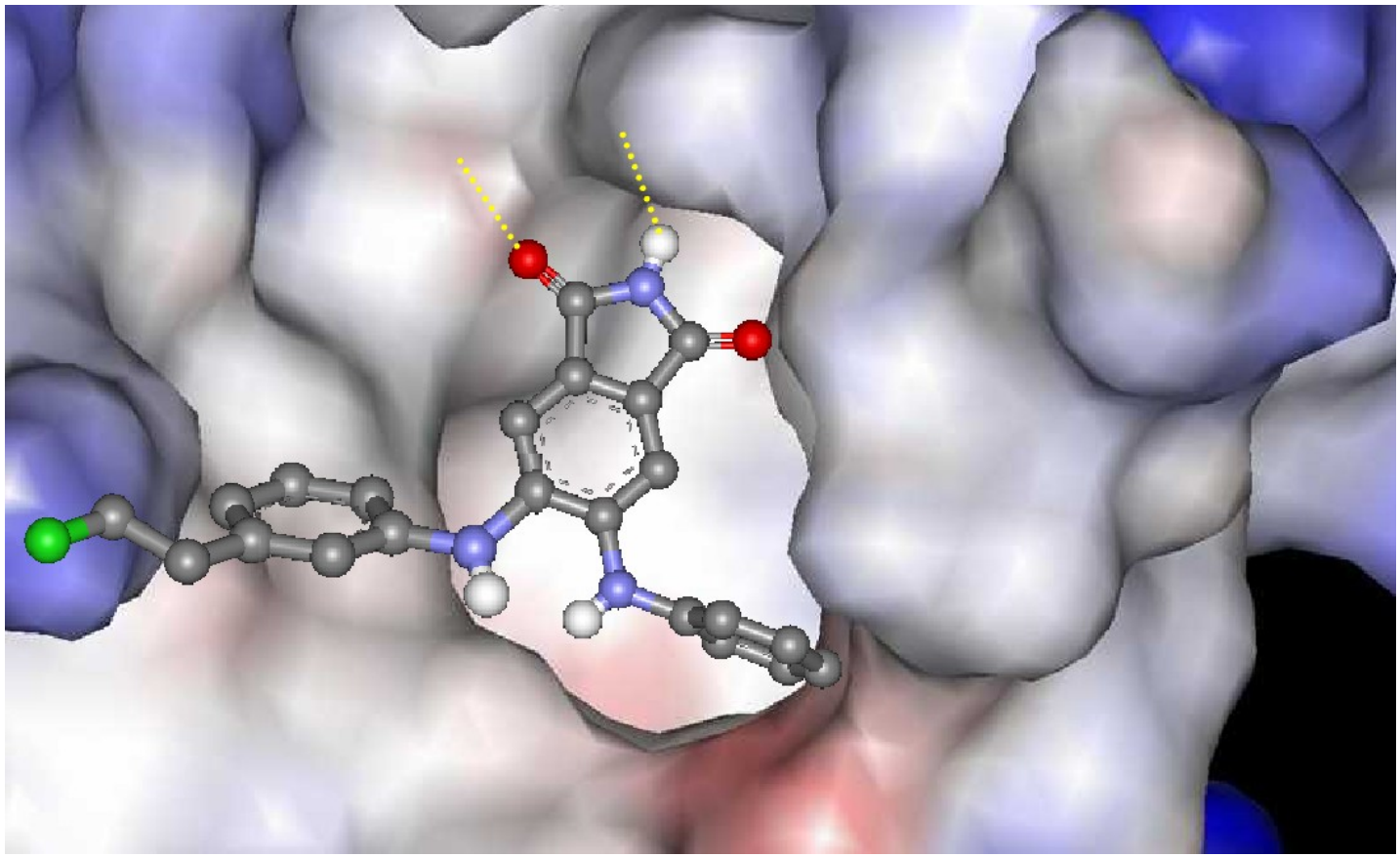
Place-and-Join Algorithm

- The place-and-join method splits the molecule into rigid subparts
- Each subpart is docked independently
- Assembly of the fragments is then done by looking at their relative location and assessing the possibility of re-connecting them with the connectivity of the initial molecule, in geometrically correct conformations



Incremental-Based Methods

- The incremental based approach starts with an initial core docked in the active site, and new fragments are progressively added and minimized; the treatment is terminated when the entire molecule is formed



Hybrid Methods

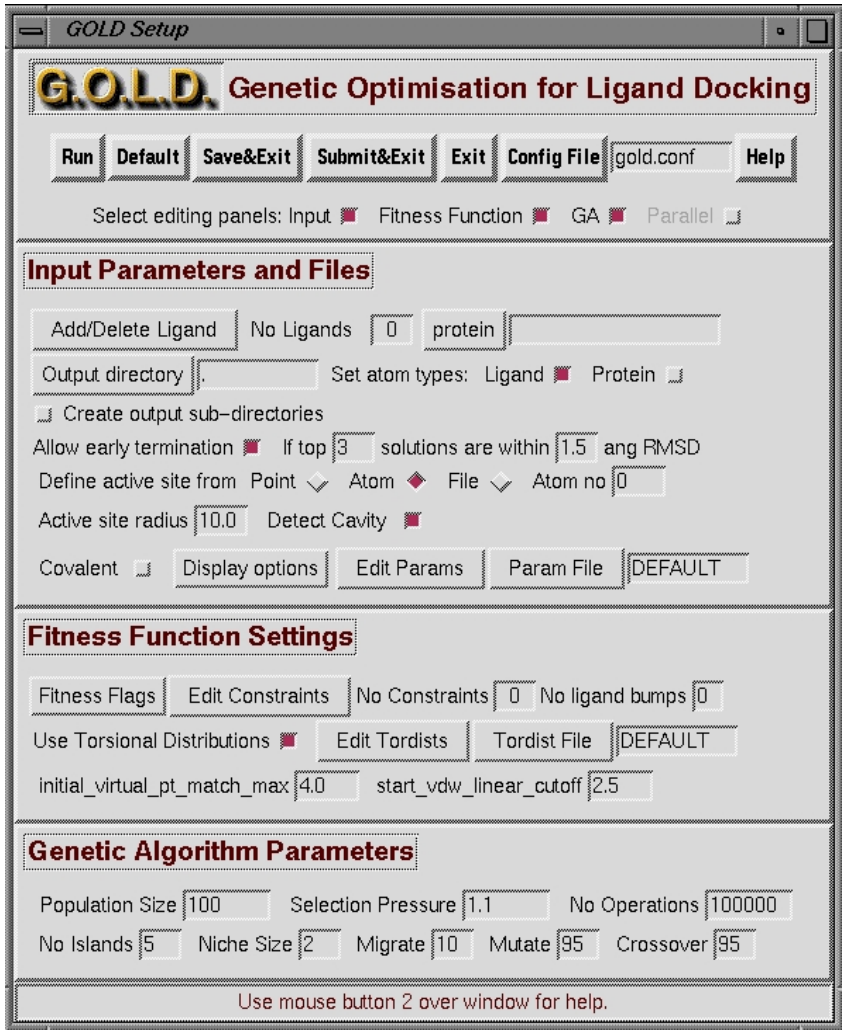
- Some of the newer methods use a hybrid scheme of a quick docking using a GA or other scheme followed by molecular dynamics to refine the prediction
- These methods are likely to be some of the best once they are correctly parameterized, etc.

GOLD

(Genetic Optimisation for Ligand Docking)

Performs automated docking with full acyclic ligand flexibility, partial cyclic ligand flexibility and partial protein flexibility in and around active site.

Scoring: includes H-bonding term, pairwise dispersion potential (hydrophobic interactions), molecular and mechanics term for internal energy.



Analysis shows algorithm more likely to fail if ligand is large or highly flexible, and more likely to succeed if ligand is polar

- The GA is encoded to search for H-bonding networks first;
- Fitness function contains a term for dispersive interactions but takes no account of desolvation, thus underestimates The Hydrophobic Effect

Scoring Functions

Scoring

- We covered some of the most common representations of the system as well as the most commonly used search methods
- All of these search methods involve evaluating the “fitness” or “energy” of a given binding conformation
- In order to do this effectively, we must have a good scoring function that can give an accurate estimate of the binding free energy (or relative free energy)

Practical Considerations

- If we have developed an efficient search algorithm, it may produce 10^9 or more potential solutions
- Many solutions can be immediately eliminated due to atomic clashes or other obvious problems, but we still must evaluate the fitness of a large number of structures
- For this reason, our scoring function must not only accurate, but it must be fast and efficient

Scoring Accuracy

- If we were scoring a single ligand-protein complex, we could adopt much more sophisticated methods to arrive at an accurate value for the binding free energy
- Requiring true accuracy in a scoring function is not a realistic expectation, however there are two features that a good scoring function should possess
- When docking a database of compounds, a good scoring function should
  - Give the best rank to the “true” bound structure
  - Give the correct relative rank of each ligand in the database
  - And again, it must be able to do these things relatively quickly

Types of Scoring Functions

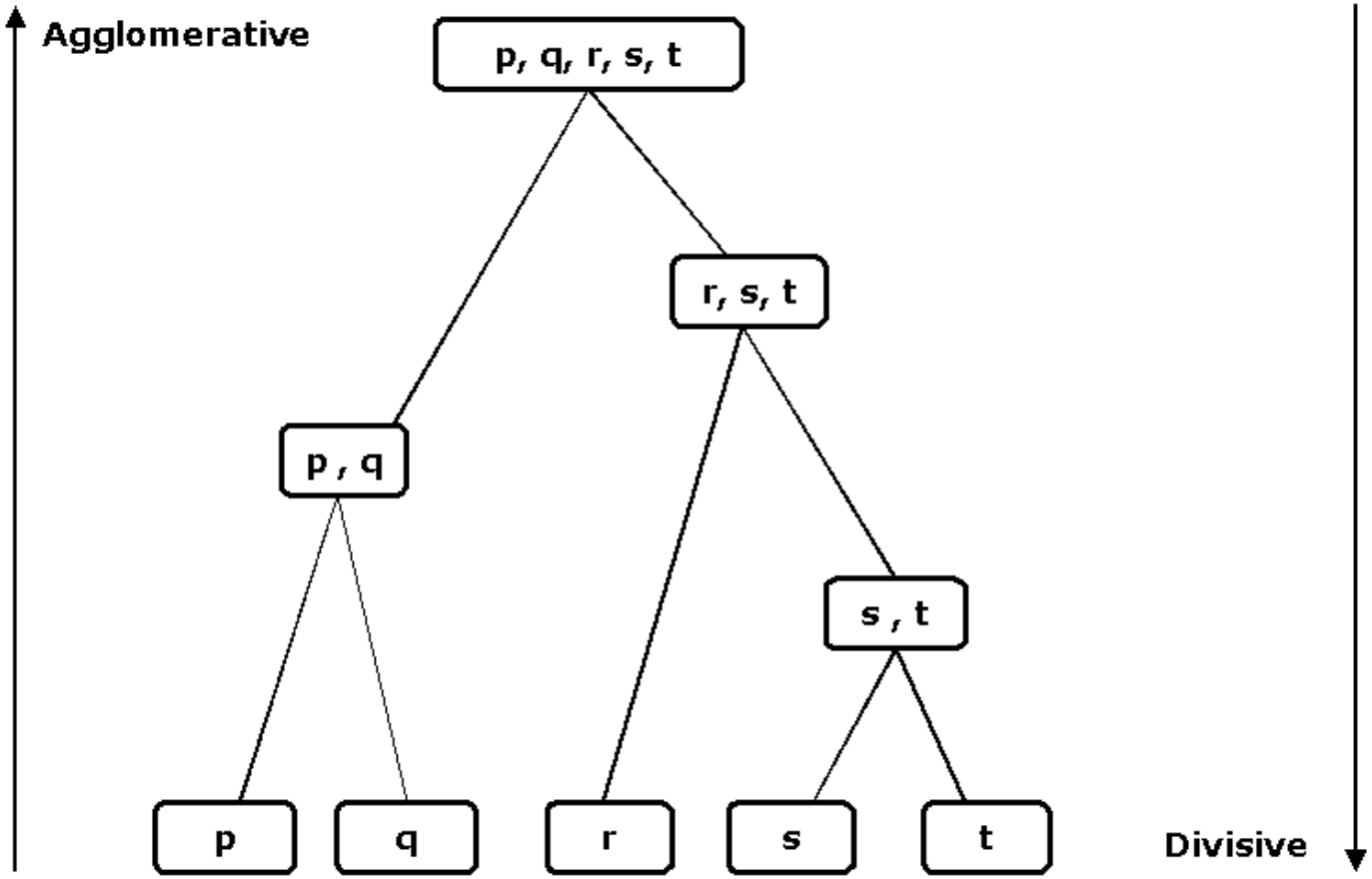
- There are several types of scoring functions that we have discussed previously
  - First Principles Methods
  - Semiempirical methods
  - Empirical methods
  - Knowledge based potentials
- All of these are used by various docking programs

Clustering

- No docking algorithm can produce a single, trustworthy structure for the bound complex, but instead they produce an ensemble of predictions
- Each predicted structure has an associated energy, but we need to consider both the binding free energy (or enthalpy as the case may be) as well as the relative population
- By clustering our data based on some “distance” criteria, we can gain some sense of similarity between predictions
- The distance can be any of a number of similarity measures, but for 3D structures, RMSD is the standard choice

Clustering

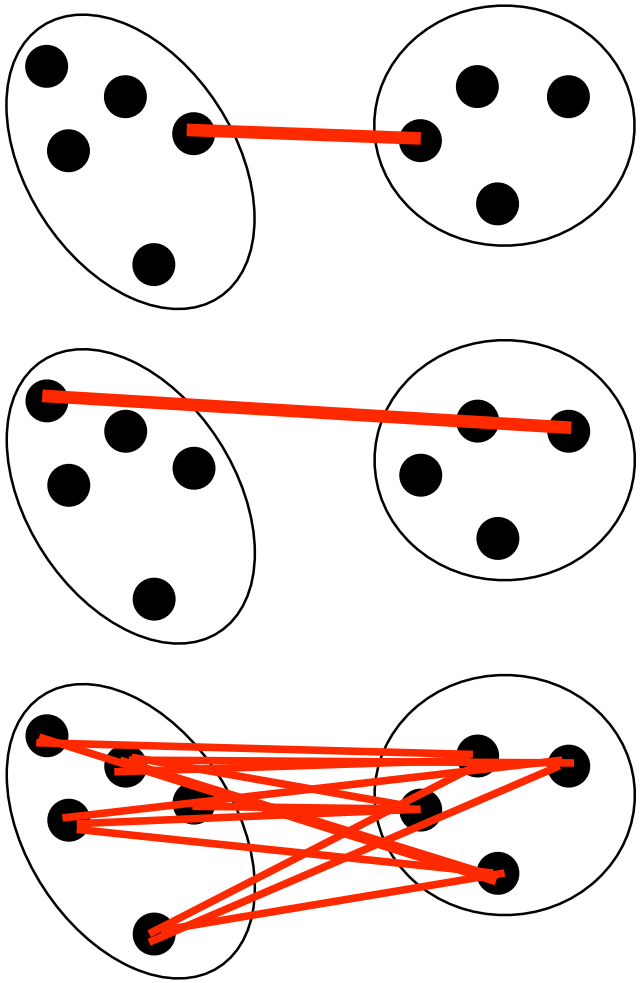
- One of the more commonly used methods is hierarchical clustering



Agglomerative Clustering

- In the agglomerative hierarchical clustering method, all structures begin as individual clusters, and the closest clusters are subsequently (iteratively) merged together
- There are again several different methods of measuring the distance between clusters
  - Simple linkage
  - Complete linkage
  - Group average

Linkage Criterion



Simple linkage selects the distance as the gives the minimum distance between any two members

Complete linkage selects the maximum distance between any two members

Group average approaches vary in effectiveness depending on the nature of the data

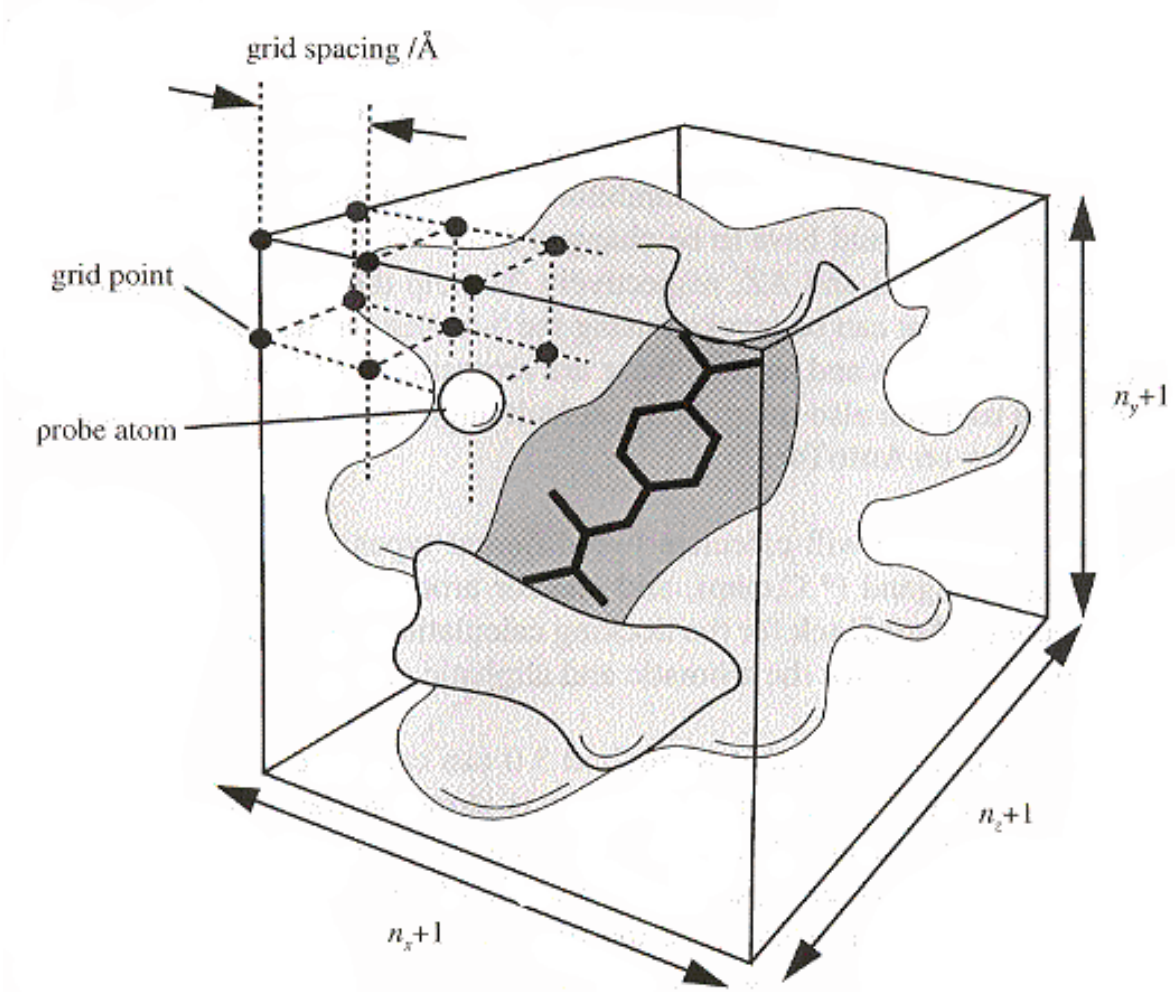
Example: Autodock

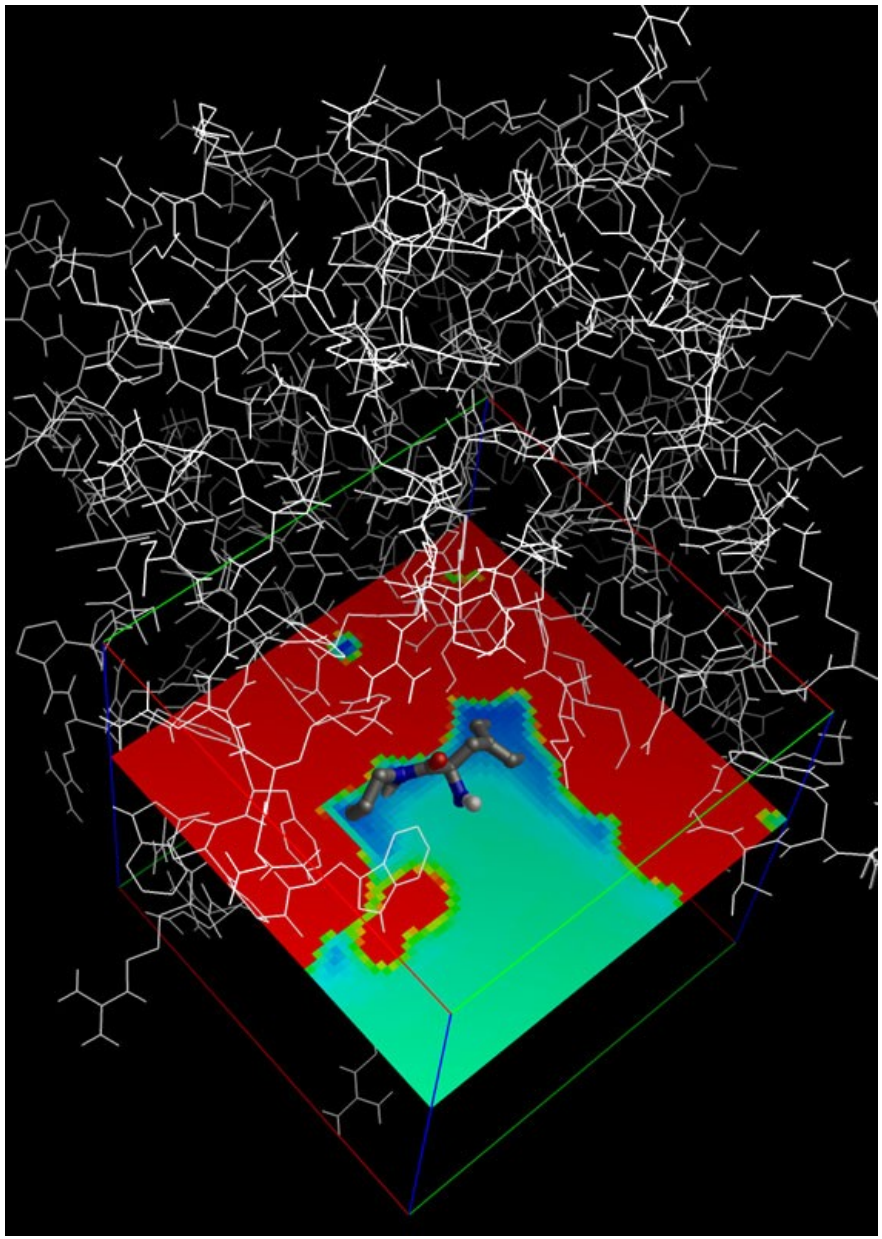
- Autodock uses pre-calculated affinity maps for each atom type in the substrate molecule, usually C, N, O and H, plus an electrostatic map
- These grids include energetic contributions from all the usual sources

$$\begin{aligned}\Delta G &= G_1 \sum_{i,j} \left(\frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}} \right) + G_2 \sum_{i,j} \left(\frac{C_{ij}}{r_{ij}^{12}} - \frac{D_{ij}}{r_{ij}^{10}} + E_{hbond} \right) \\ &+ G_3 \sum_{i,j} \frac{q_i q_j}{\epsilon(r) r_{ij}} + G_4 \Delta G_{tor} + G_5 \sum_{i,j} S_i V_j e^{-r_{ij}^2/2\sigma^2}\end{aligned}$$

Grid Maps

Each type of atom is placed at each individual grid point and the change in free energy is calculated





Grid Maps

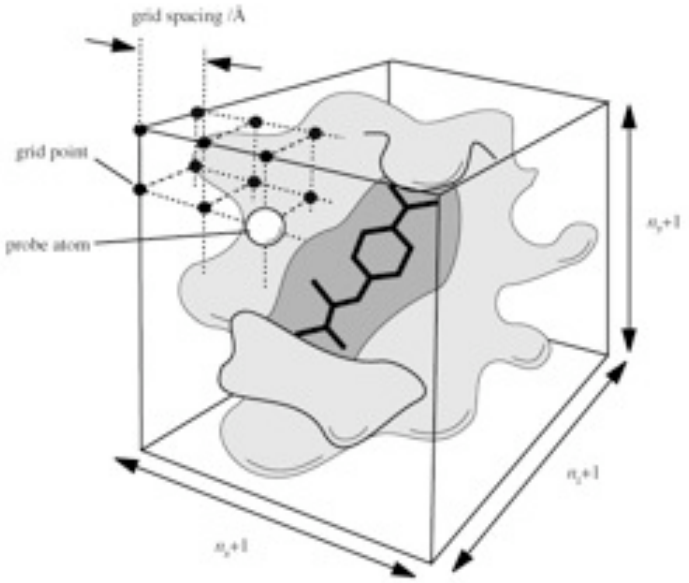
Precompute interactions for each type of atom

100X faster than pairwise methods

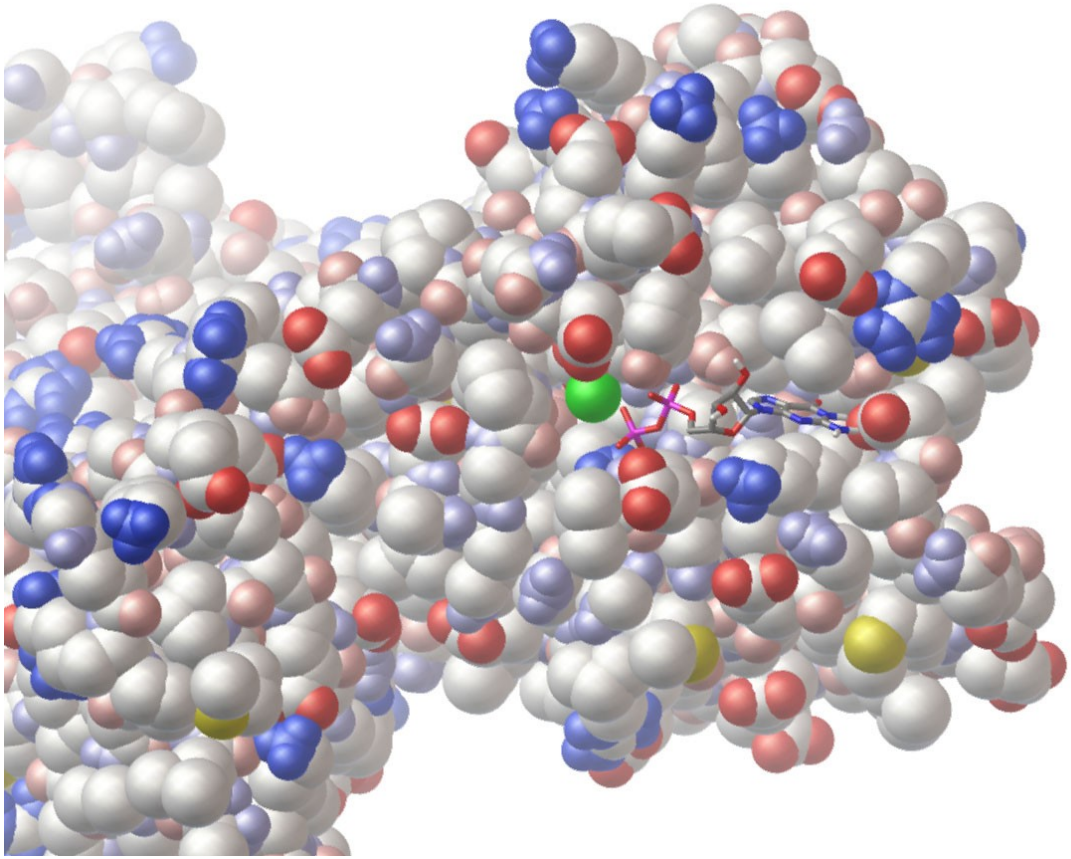
Drawbacks: receptor is conformationally rigid, limits the search space

Setting up the AutoGrid Box

- Macromolecule atoms in the rigid part
- Center:
  - center of ligand;
  - center of macromolecule;
  - a picked atom; or
  - typed-in x-, y- and z-coordinates.
- Grid point spacing:
  - default is 0.375Å (from 0.2Å to 1.0Å:).
- Number of grid points in each dimension:
  - only give even numbers (from 2 × 2 × 2 to 126 × 126 × 126).
  - AutoGrid adds one point to each dimension.
- Grid Maps depend on the orientation of the macromolecule.

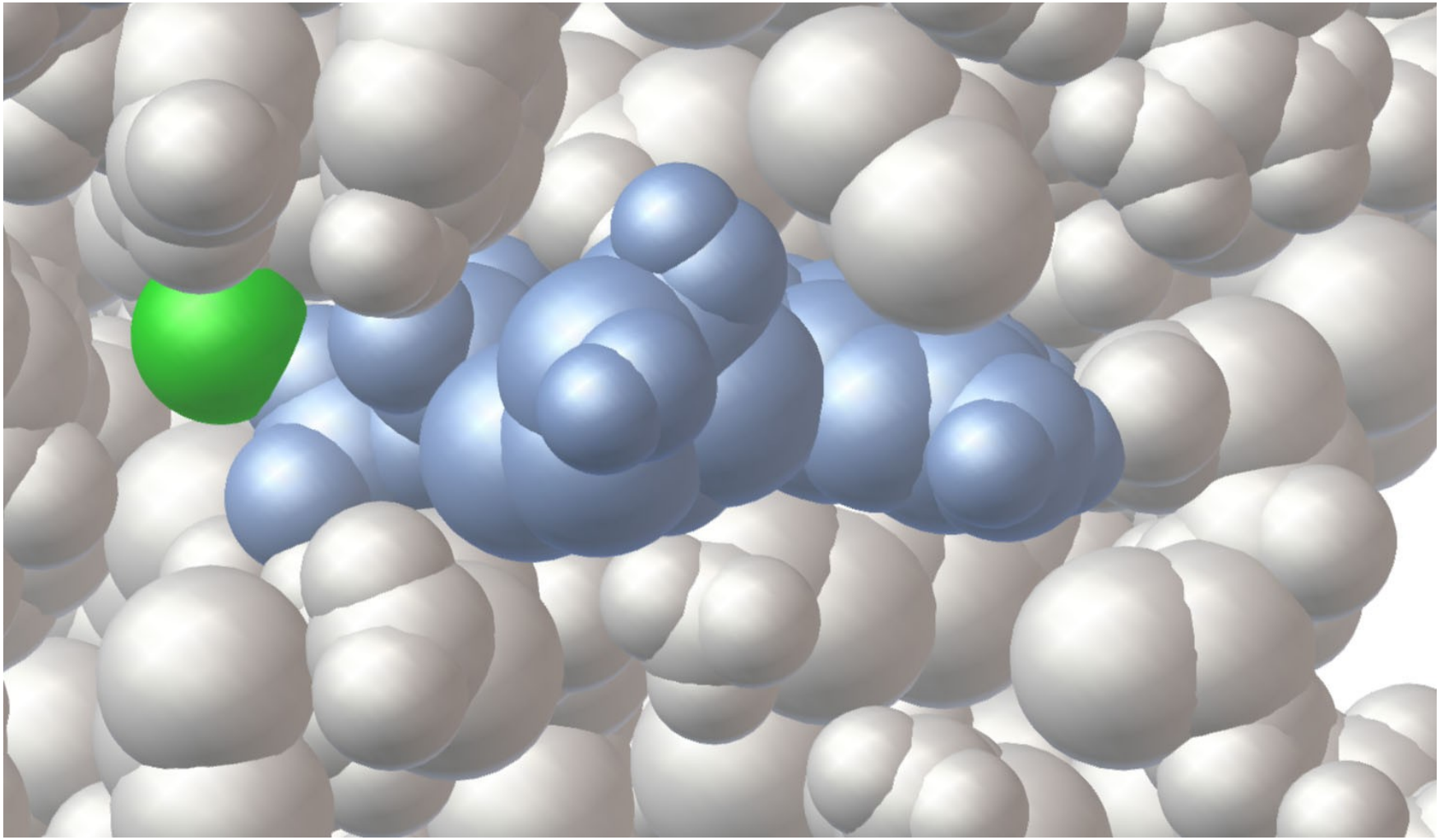


Scoring Functions



$$\Delta G_{binding} = \Delta G_{vdW} + \Delta G_{elec} + \Delta G_{hbond} + \Delta G_{desolv} + \Delta G_{tors}$$

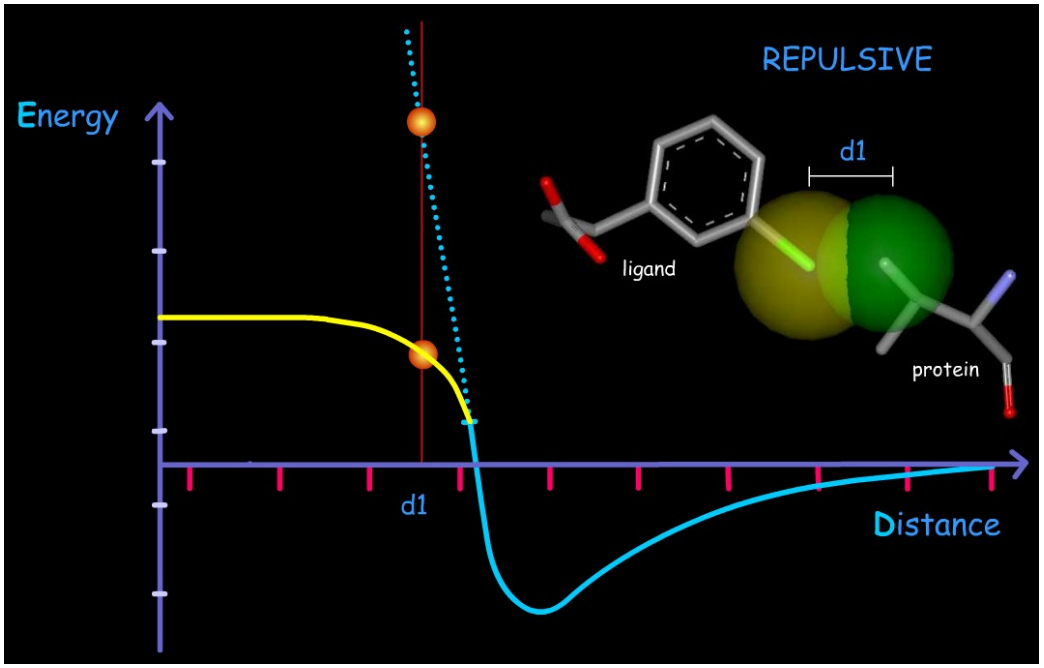
Dispersion/Repulsion



$$\Delta G_{binding} = \Delta G_{vdW} + \Delta G_{elec} + \Delta G_{hbond} + \Delta G_{desolv} + \Delta G_{tors}$$

Soft Van der Waals Repulsion Functions

- When calculated with force fields from molecular mechanics, steric clashes correspond to high energies
- Modifying the normal Van der Waals repulsion function (in blue) into a softer curve (such as the yellow one) enables them to be less dominant and to simulate the plasticity of the receptor without changing its geometry



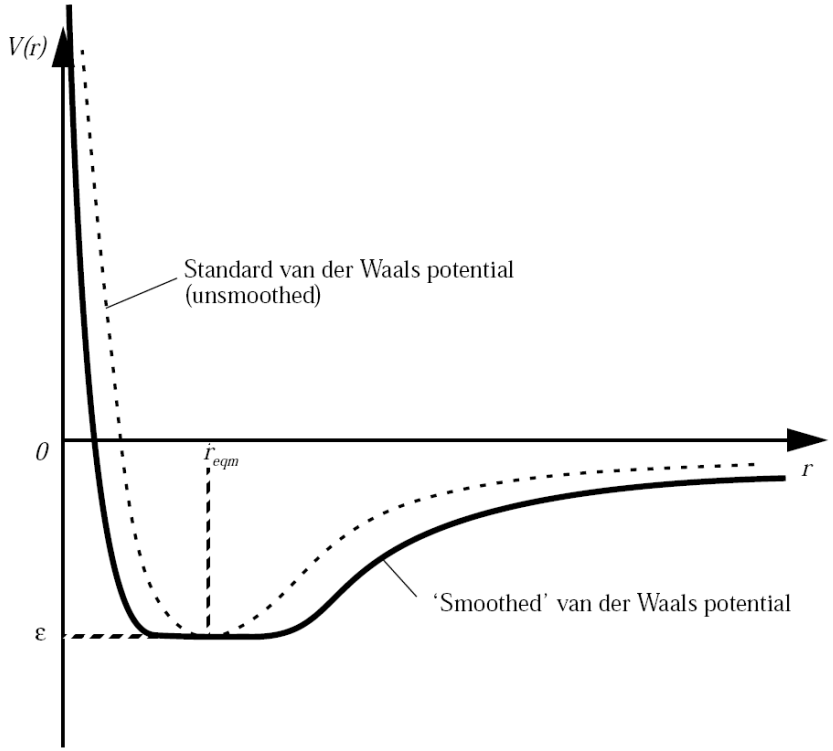
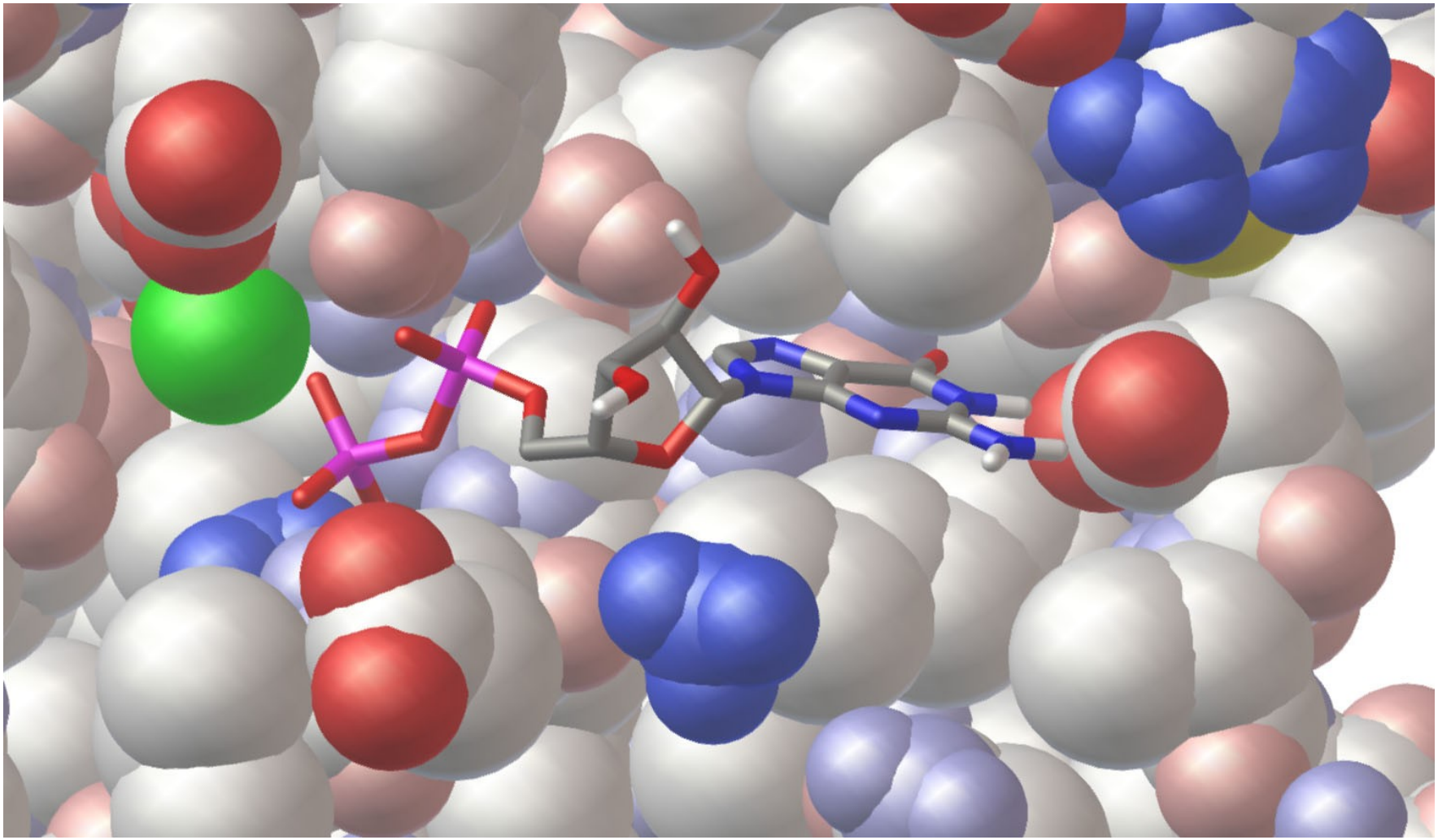


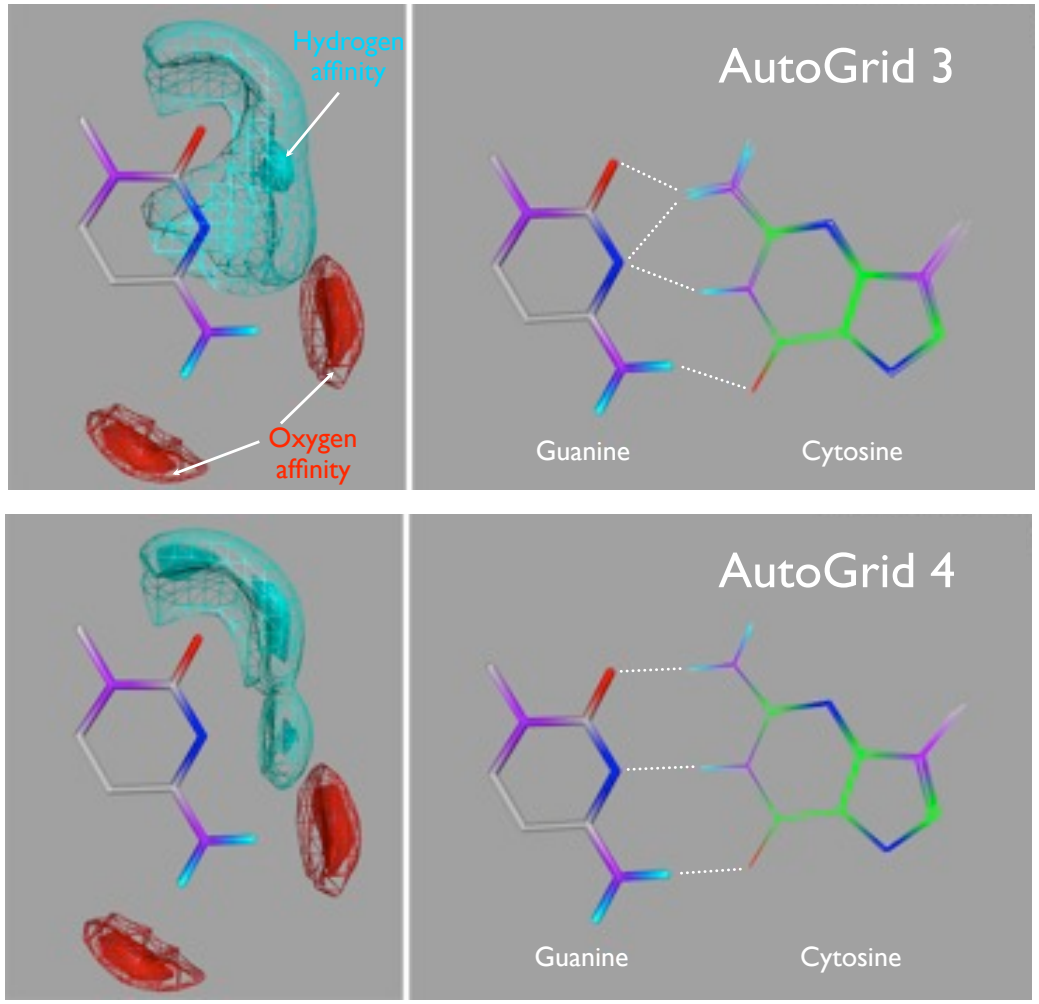
Fig. 2: Another approach implemented in AutoDock software

Electrostatics and Hydrogen Bonds



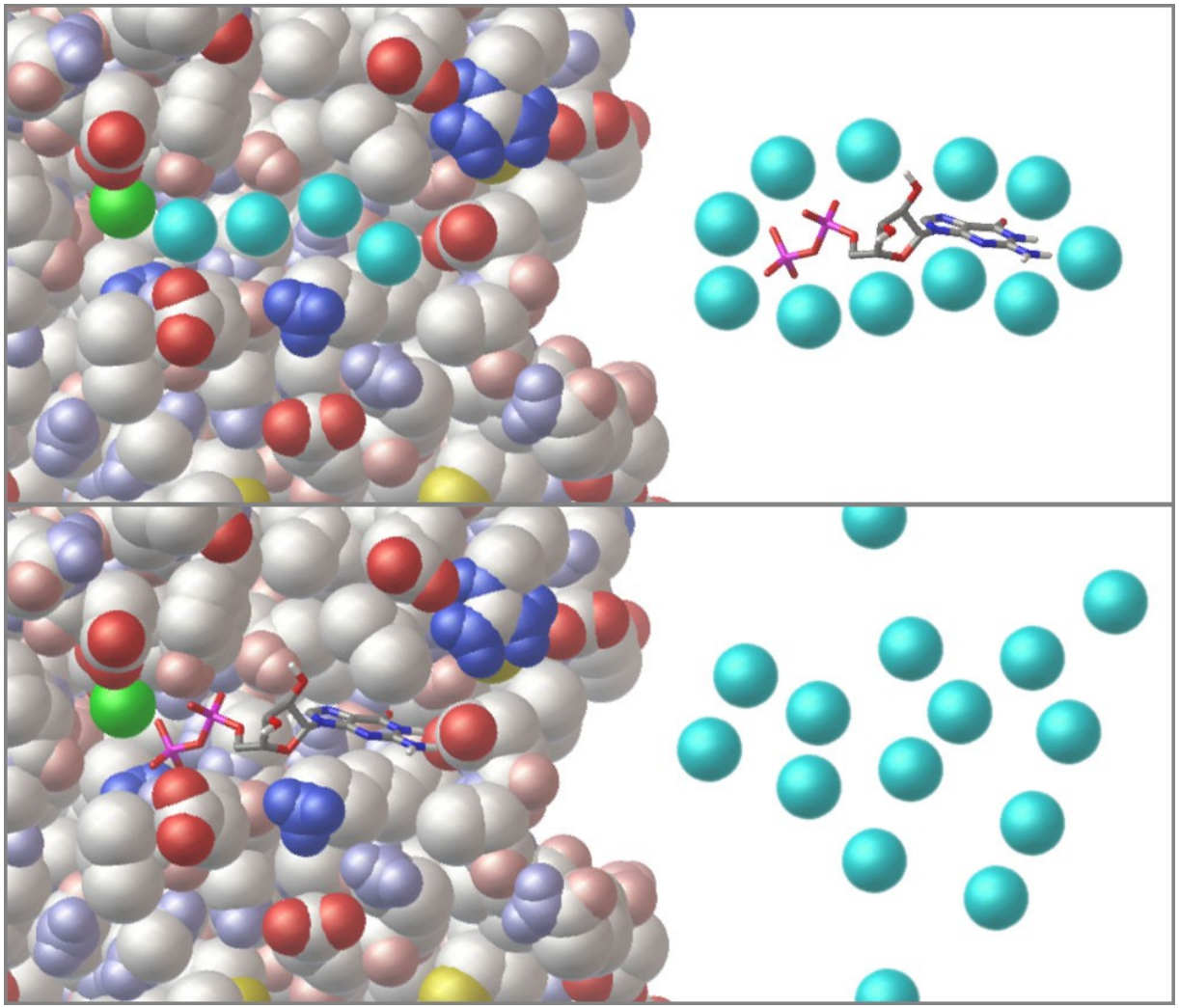
$$\Delta G_{binding} = \Delta G_{vdW} + \Delta G_{elec} + \Delta G_{hbond} + \Delta G_{desolv} + \Delta G_{tors}$$

Improved H-bond Directionality



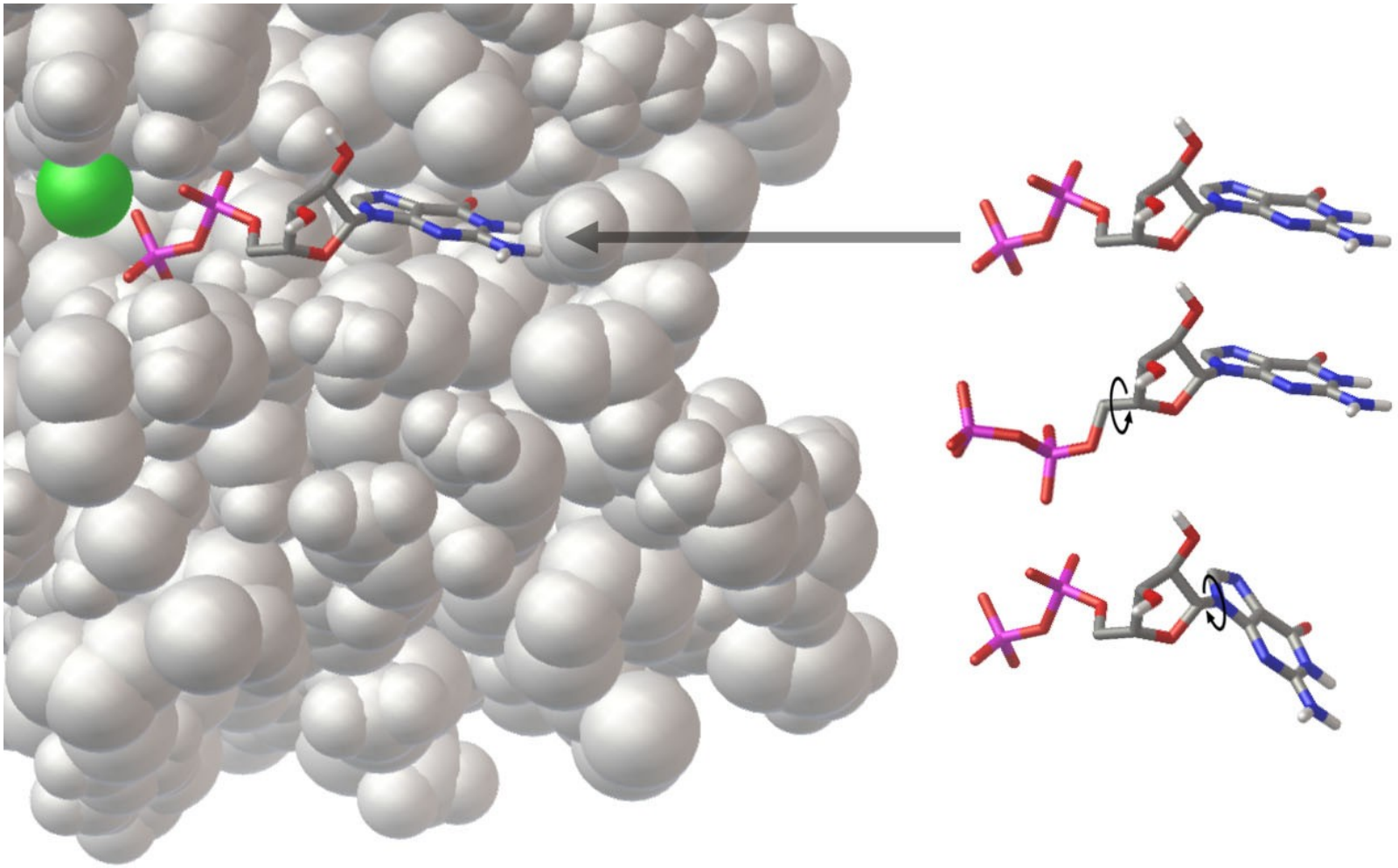
Huey, Goodsell, Morris, and Olson (2004) Letts. Drug Des. & Disc., 1: 178-183

Desolvation



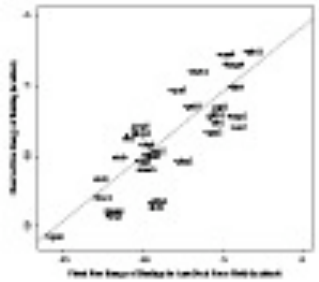
$$\Delta G_{binding} = \Delta G_{vdW} + \Delta G_{elec} + \Delta G_{hbond} + \Delta G_{desolv} + \Delta G_{tors}$$

Torsional Entropy



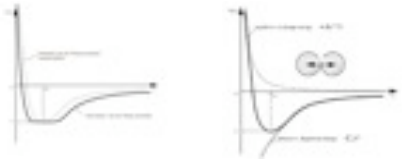
$$\Delta G_{binding} = \Delta G_{vdW} + \Delta G_{elec} + \Delta G_{hbond} + \Delta G_{desolv} + \Delta G_{tors}$$

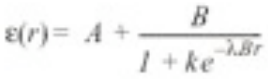
AutoDock4 Scoring Function Terms

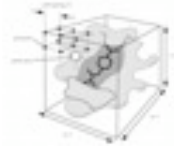


$$\Delta G_{binding} = \Delta G_{vdW} + \Delta G_{elec} + \Delta G_{hbond} + \Delta G_{desolv} + \Delta g_{tors}$$

http://autodock.scripps.edu/science/equations
http://autodock.scripps.edu/science/autodock-4-desolvation-free-energy/







- $\Delta G_{vdW} = \Delta G_{vdW}$

  12-6 Lennard-Jones potential (with 0.5 Å smoothing)

- ΔG_{elec}

  with Solmajer & Mehler distance-dependent dielectric

- ΔG_{hbond}

  12-10 H-bonding Potential with Goodford Directionality

- ΔG_{desolv}

  Charge-dependent variant of Stouten Pairwise Atomic Solvation Parameters

- ΔG_{tors}

  Number of rotatable bonds

AutoDock Empirical Free Energy Force Field

$$W_{vdw}\sum_{i,j}\left(\frac{A_{ij}}{r_{ij}^{12}}-\frac{B_{ij}}{r_{ij}^{6}}\right)+$$

$$W_{hbond}\sum_{i,j}E(t)\left(\frac{C_{ij}}{r_{ij}^{12}}-\frac{D_{ij}}{r_{ij}^{10}}\right)+$$

$$W_{elec}\sum_{i,j}\frac{q_i q_j}{\varepsilon(r_{ij})r_{ij}}+$$

$$W_{sol}\sum_{i,j}(S_i V_j + S_j V_i)e^{(-r_{ij}^2/2\sigma^2)}+$$

$$W_{tor}N_{tor}$$

Physics-based approach from molecular mechanics

Calibrated with 188 complexes from LPDB, K_i*'s* from PDB-Bind

Standard error = 2.52 kcal/mol

Scoring Function in AutoDock 4

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

- Desolvation includes terms for all atom types
  - Favorable term for C, A (aliphatic and aromatic carbons)
  - Unfavorable term for O, N
  - Proportional to the absolute value of the charge on the atom
  - Computes the intramolecular desolvation energy for moving atoms
- Calibrated with 188 complexes from LPDB, K_i's from PDB-Bind

  Standard error (in Kcal/mol):
  - 2.62 (extended)
  - 2.72 (compact)
  - 2.52 (bound)
  - 2.63 (AutoDock 3, bound)
- Improved H-bond directionality

AutoDock Vina Scoring Function

Combination of knowledge-based and empirical approach

$$\Delta G_{binding} = \Delta G_{gauss} + \Delta G_{repulsion} + \Delta G_{hbond} + \Delta G_{hydrophobic} + \Delta G_{tors}$$

ΔG_{gauss}

Attractive term for dispersion, two gaussian functions

$\Delta G_{repulsion}$

Square of the distance if closer than a threshold value

ΔG_{hbond}

Ramp function - also used for interactions with metal ions

$\Delta G_{hydrophobic}$

Ramp function

ΔG_{tors}

Proportional to the number of rotatable bonds

Calibrated with 1,300 complexes from PDB-Bind

Standard error = 2.85 kcal/mol

Other Energetic Notes

- Also includes torsional energy (if the ligand is flexible)
- Entropic contributions due to rotatable bonds
- Desolvation is calculated, but only for heavy atoms on the marcromolecule

Search Methods

- Autodock has three search methods
  - Monte Carlo simulated annealing
  - A global genetic algorithm search
  - A local Solis and Wets search
- Combining the last two search methods gives the Lamarckian Genetic Algorithm which is what we typically use

AutoDock and Vina Search Methods

Global search algorithms:

- Simulated Annealing (Goodsell *et al.* 1990)
- Genetic Algorithm (Morris *et al.* 1998)

Local search algorithm:

- Solis & Wets (Morris *et al.* 1998)

Hybrid global-local search algorithm:

- Lamarckian GA (Morris *et al.* 1998)

Iterated Local Search:

- Genetic Algorithm with Local Gradient Optimization (Trott and Olson 2010)

Methodology

- Require structures for both ligand and macromolecule
- Add charges to both structures (create a *pdbq* file)
  - For proteins or polypeptides we use the Kollman united atom model
  - For drugs and other molecule, we use Gasteiger charges

Methodology

- For the macromolecule, solvation needs to be calculated (producing a *pdbqs* file)
- Based on the atom types in the ligand, the appropriate maps need to be calculated for the macromolecule
- Flexible bonds (if any) need to be assigned to the ligand

Practical Considerations

- What problems are feasible?
  - Depends on the search method:
    - Vina > LGA > GA >> SA >> LS
    - AutoDock SA : can output trajectories, $D < 8$ torsions.
    - AutoDock LGA : $D <$ 8-16 torsions.
    - Vina : good for 20-30 torsions.
- When are AutoDock and Vina not suitable?
  - Modeled structure of poor quality;
  - Too many torsions (32 max);
  - Target protein too flexible.
- Redocking studies are used to validate the method

Using AutoDock: Step-by-Step

- Set up ligand PDBQT—using ADT's "Ligand" menu
- OPTIONAL: Set up flexible receptor PDBQT—using ADT's "Flexible Residues" menu
- Set up macromolecule & grid maps—using ADT's "Grid" menu
- Pre-compute AutoGrid maps for all atom types in your set of ligands—using "`autogrid4`"
- Perform dockings of ligand to target—using "`autodock4`", and in parallel if possible.
- Visualize AutoDock results—using ADT's "Analyze" menu
- Cluster dockings—using "analysis" DPF command in "autodock4" or ADT's "Analyze" menu for parallel docking results.

AutoDock 4 File Formats

Prepare the Following Input Files

- Ligand PDBQT file
- Rigid Macromolecule PDBQT file
- Flexible Macromolecule PDBQT file ("Flexres")
- AutoGrid Parameter File (GPF)
  - GPF depends on atom types in:
    - Ligand PDBQT file
    - Optional flexible residue PDBQT files)
- AutoDock Parameter File (DPF)

Run AutoGrid 4

- Macromolecule PDBQT + GPF → Grid Maps, GLG

Run AutoDock 4

- Grid Maps + Ligand PDBQT [+ Flexres PDBQT] + DPF → DLG (dockings & clustering)

Things you need to do before using AutoDock 4

Ligand:

- Add all hydrogens, compute Gasteiger charges, and merge non-polar H; also assign AutoDock 4 atom types
- Ensure total charge corresponds to tautomeric state
- Choose torsion tree root & rotatable bonds

Macromolecule:

- Add all hydrogens (PDB2PQR flips Asn, Gln, His), compute Gasteiger charges, and merge non-polar H; also assign AutoDock 4 atom types
- Assign Stouten atomic solvation parameters
- Optionally, create a flexible residues PDBQT in addition to the rigid PDBQT file
- Compute AutoGrid maps

Preparing Ligands and Receptors

- AutoDock uses 'United Atom' model
  - Reduces number of atoms, speeds up docking
- Need to:
  - Add polar Hs. Remove non-polar Hs.
    - Both Ligand & Macromolecule
  - Replace missing atoms (disorder).
  - Fix hydrogens at chain breaks.
- Need to consider pH:
  - Acidic & Basic residues, Histidines.
  - http://molprobity.biochem.duke.edu/
- Other molecules in receptor:
    - Waters; Cofactors; Metal ions.
- Molecular Modelling elsewhere.

Atom Types in AutoDock 4

- One-letter or two-letter atom type codes
- More atom types than AD3:
  - 22
- Same atom types in both ligand and receptor
- http://autodock.scripps.edu/wiki/NewFeatures
- http://autodock.scripps.edu/faqs-help/faq/how-do-i-add-new-atom-types-to-autodock-4
- http://autodock.scripps.edu/faqs-help/faq/where-do-i-set-the-autodock-4-force-field-parameters

Partial Atomic Charges are required for both Ligand and Receptor

- Partial Atomic Charges:
  - Peptides & Proteins; DNA & RNA
    - Gasteiger (PEOE) - AD_4 Force Field
  - Organic compounds; Cofactors
    - Gasteiger (PEOE) - AD_4 Force Field;
    - MOPAC (MNDO, AM_1, PM_3);
    - Gaussian (6-31G*).
- Integer total charge per residue.
- Non-polar hydrogens:
  - Always merge

Carbon Atoms can be either Aliphatic or Aromatic Atom Types

- Solvation Free Energy
  - Based on a partial-charge-dependent variant of Stouten method.
  - Treats aliphatic ('C') and aromatic ('A') carbons differently.
- Need to rename ligand aromatic 'C' to 'A'.
- ADT determines if ligand is a peptide:
  - If so, uses a look-up dictionary.
  - If not, inspects geometry of 'C's in rings. Renames 'C' to 'A' if flat enough.
  - Can adjust 'planarity' criterion (15° detects more rings than default 7.5°).

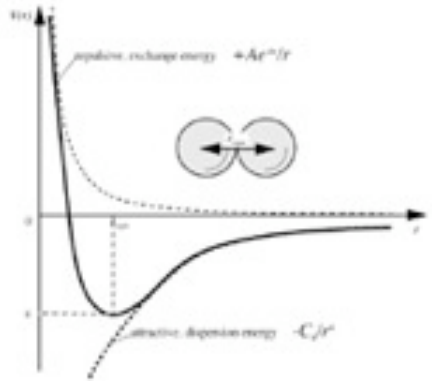
Defining Ligand Flexibility

- Set Root of Torsion Tree:
  - By interactively picking, or
  - Automatically.
    - Smallest 'largest sub-tree'.
- Interactively Pick Rotatable Bonds:
  - No 'leaves';
  - No bonds in rings;
  - Can freeze:
    - Peptide/amide/selected/all;
  - Can set the number of active torsions that move either the most or the fewest atoms

Setting Up Your Environment

- At TSRI:
  - Modify .cshrc
    - Change PATH & stacksize:
    - setenv PATH (/mgl/prog/$archosv/bin:/tsri/python:$path)
    - % limit stacksize unlimited
- ADT Tutorial, every time you open a Shell or Terminal, type:
    - % source /tsri/python/share/bin/initadtcsh
- To start AutoDockTools (ADT), type:
  - % cd tutorial
  - % adt1
- Web
  - http://autodock.scripps.edu

Choose the Docking Algorithm



- SA.dpf – Simulated Annealing
- GA.dpf – Genetic Algorithm
- LS.dpf – Local Search
  - Solis-Wets (SW)
  - Pseudo Solis-Wets (pSW)
- GALS.dpf – Genetic Algorithm with Local Search, i.e. Lamarckian GA

Run AutoGrid

- Check: Enough disk space?
  - Maps are ASCII, but can be ~2-8MB !
- Start AutoGrid from the Shell:

```
% autogrid4 –p mol.gpf –l mol.glg &
% autogrid4 -p mol.gpf -l mol.glg ; autodock4 -p mol.dpf -l mol.dlg
```

- Follow the log file using:

```
% tail -f mol.glg
```

  - Type <Ctrl>-C to break out of the 'tail -f' command
- Wait for "Successful Completion" before starting AutoDock

Run AutoDock

- Do a test docking, ~ 25,000 evals
- Do a full docking, if test is OK, ~ 250,000 to 50,000,000 evals
- From the Shell:
  - `% autodock4 –p yourFile.dpf –l yourFile.dlg &`
- Expected time? Size of docking log?
- Distributed computation
  - At TSRI, Linux Clusters
  - ```
    % submit.py stem 20
    % recluster.py stem 20 during 3.5
    ```
    ```

Analyzing AutoDock Results

- In ADT, you can:
  - Read & view a single DLG, or
  - Read & view many DLG results files in a single directory
  - Re-cluster docking results by conformation & view these
- Outside ADT, you can re-cluster several DLGs
  - Useful in distributed docking
    - `% recluster.py stem 20 [during|end] 3.5`

Viewing Conformational Clusters by RMSD

- List the RMSD tolerances
  - Separated by spaces
- Histogram of conformational clusters
  - Number in cluster versus lowest energy in that cluster
- Picking a cluster
  - makes a list of the conformations in that cluster;
  - set these to be the current sequence for states player.