Lecture Notes for 20.320 Fall 2012

Molecular Design Part 1

Ernest Fraenkel

In previous lectures we have examined the relationship of structure to energy and the basis for specificity in protein-ligand interactions. We will now see how the two topics can be brought together for the purpose of design. We will explore the techniques needed to design mutations that can alter specificity. Very similar methods were used to design the mutant GCSF with increased potency that we described in the introduction to this section of the course. Finally, we will examine how computation analysis of non-protein ligands can aid in drug-discovery.

A Crisis in the Pharmaceutical Industry.

The pharmaceutical industry gross income is approximately $600 billion each year and companies pour lots of money back into research. (I've seen estimates as high as 20% of sales). Nevertheless, in the last few years there have only been about 20 new chemical entities approved as drugs. At this rate, the industry is becoming unsustainable.

Why are so few new drugs invented each year? The main problem is not at the level of initial discovery, but the high rate of attrition.

- Only one in 5,000 compounds discovered in pharma research ever gets to a clinical trial.
- Of the compounds that are tested, only about 1/15 becomes a commercial drug.

Can new technologies changes these numbers? Genomics and systems biology are increasing our ability to find the molecular origin for diseases. These techniques can help identify good targets, but the down-stream steps in drug discovery are still slow and yield the high rates of failure. In this section we will explore how computational techniques can help in predicting the interaction of potential drugs with on-target and off-target proteins, and how these methods could dramatically improve drug discovery.

In the first part of this topic, we will learn how to predict the effect an amino-acid mutation will have on the interaction of a protein ligand with its target. In the second part we will explore non-protein ligands.

Part 1: Designing Mutations in Protein Ligands.

Overview of this topic:

1. We are interested in predicting the free energy change for a mutated complex. Using thermodynamic cycles allows us to identify "alchemical transitions" from which it is possible to estimate the energy change.

2. To compute the energy of these transitions we need to repack the protein's side chains to account for the mutation.

3. We convert this problem into a combinatorial problem by using the concept of "rotamers."

4. We apply a sampling technique called the metropolis algorithm to solve this extremely large combinatorial search problem.

1. Predicting the free energy change of a complex.

Let's consider a relatively simple case: we want to find a mutation in a protein ligand that increases its affinity for a receptor. Let's start by writing the equilibrium reactions:

$R + L \Leftrightarrow C$ has a free energy of $\Delta G^{wt}_{,bind}$

$R + Lm \Leftrightarrow C^*$ has a free energy of $\Delta G^{mut}_{, bind}$

Notice the notation: the superscript refers to the state that isn't changing in the reaction (wt vs. mutant) and the subscript refers to the reaction.

We define $\Delta\Delta G = \Delta G^{mut}_{, bind} - \Delta G^{wt}_{,bind}$

If we can predict how a mutation affects $\Delta\Delta G$ then we can find a mutation that increases affinity. How can we predict this value? We cannot simulate the binding reaction for each protein because of the difficulty in modeling solvation and conformational changes.

The solution to the problem comes from the realization that free energy is a state function, and thus is independent of path. So let's re-examine our problem and consider the transition from the wild-type to the mutant ligand. This transformation

isn't exactly a chemical reaction, it doesn't conserve mass. This is frequently called an alchemical reaction – a reference to alchemy.

$$\Delta G^{wt}_{,bind}$$

$$R + L \Leftrightarrow C$$

$$\updownarrow \qquad \updownarrow$$

$$R + L_m \Leftrightarrow C^*$$

$$\Delta G^{mut}_{,bind}$$

"The Alchemist" by Pieter Bruegel the Elder

The horizontal reactions are the
physically reasonable ones, but they are hard to evaluate computationally because they depend on solvent effects, structural changes, etc. The vertical transitions couldn't happen in reality, but we will see that they are very easy to compute – under certain assumptions.

So we can assign a free energy to the vertical transitions as well:

$$\Delta G^{wt}_{,bind}$$

$$R + L \Leftrightarrow C$$

$$\Delta G^{unb}_{,mutate} \quad \updownarrow \qquad \updownarrow \quad \Delta G^{bnd}_{,mutate}$$

$$R + L_m \Leftrightarrow C^*$$

$$\Delta G^{mut}_{,bind}$$

Because free energy is a state function, it has to sum to zero if I go around this cycle. So we get $\Delta G^{wt}_{,bind} + \Delta G^{bnd}_{,mutate} - \Delta G^{mut}_{,bind} - \Delta G^{unb}_{,mutate} = 0$

or $\Delta G^{bnd}_{,mutate} - \Delta G^{unb}_{,mutate} = \Delta G^{mut}_{,bind} - \Delta G^{wt}_{,bind}$

Now recall that $\Delta\Delta G = \Delta G^{mut}_{,bind} - \Delta G^{wt}_{,bind}$

So, there are two ways to calculate $\Delta\Delta G$, the transition we can simulate, if our assumptions are valid, and the experimental direction that we can measure but not simulate.

This suggests that we can figure out the change in affinity of our mutation by computing the energy of the two alchemical reactions – the kind that is easiest for the computer to solve.

Under the assumption that the overall structure is the same in the wild-type and the mutant complex our procedure can be relatively simple.

Each term in $\Delta\Delta G = \Delta G^{bnd}_{,\,mutate} - \Delta G^{unb}_{,\,mutate}$ corresponds to two structures in which all the atoms are the same except for the side-chain of the mutated amino acid. Let's assume we are dealing with a Alanine to Glycine residue.

What atoms are changing?

Do we need to compute $\Delta G^{unb}_{,\,mutate}$? Yes, as there can be differences in the energy of the wt and mutant protein even in isolation. These differences can arise from loss of van der Waals interactions with other atoms in the ligand and changes in solvation.

We can compute $\Delta G^{unb}_{,\,mutate}$ by relating it to the potential energy functions we looked at in previous lectures:

$$\Delta G^{unb}_{,\,mutate} = U^{unb,\,gly} - U^{unb,\,ala}$$

where U is the potential energy, which is the sum of the vdW and electrostatic terms.

Now recall that we sum over all atomic pairs, but since we are taking the difference between two potential energy functions, we only need to consider those terms that differ between the two states.

For example, if we mutate residue 10 in protein P, which has N residues, the only terms we need to consider are

$$\Delta G^{unb}_{mutate} = \sum_{i=1}^{N} U^{vdW}_{P_i,P_{G10}} + U^{elect}_{P_i,P_{G10}} - \sum_{i=1}^{N} U^{vdW}_{P_i,P_{A10}} + U^{elect}_{P_i,P_{A10}},$$

which represents the interactions of our new methyl group with all the other residues in the protein.

What are the additional terms in $\Delta G^{bnd}_{,\,mutate}$?

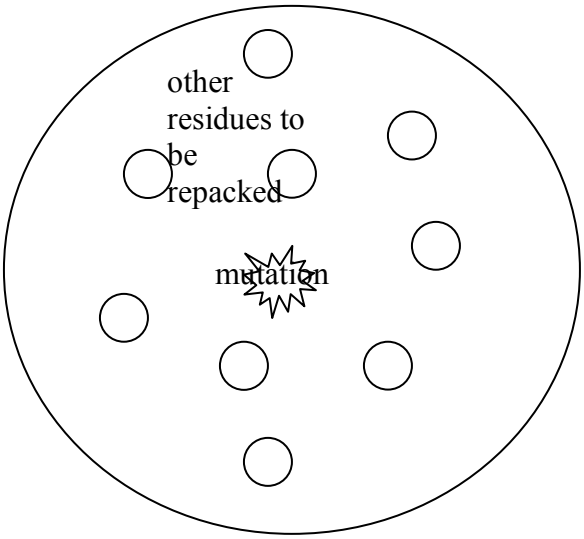
$$\Delta G^{bnd}_{mutate} = \sum_{i=1}^{N}\left[U^{vdW}_{P_i,P_{G10}} + U^{elect}_{P_i,P_{G10}}\right] + \sum_{i=1}^{M}\left[U^{vdW}_{Q_i,P_{G10}} + U^{elect}_{Q_i,P_{G10}}\right]$$
$$-\left[\sum_{i=1}^{N}\left[U^{vdW}_{P_i,P_{A10}} + U^{elect}_{P_i,P_{A10}}\right] + \sum_{i=1}^{M}\left[U^{vdW}_{Q_i,P_{A10}} + U^{elect}_{Q_i,P_{A10}}\right]\right]$$

(where the interacting ligand has M residues).

2. Repacking the protein's side chains.

What happens if we replace a small amino acid such as alanine with a larger, charged amino acid, say lysine? The new residue needs to go somewhere, and presumably the rest of the structure has to change to accommodate it. So there are now two problems. We need to recompute the energy terms we saw before, but only after we figure out where the new atoms go. Even if there is some space into which we could put the side chain without introducing any steric clashes, we cannot assume that that position is OK. Instead, we need to find the conformation that minimizes of the free energy for the mutated protein.

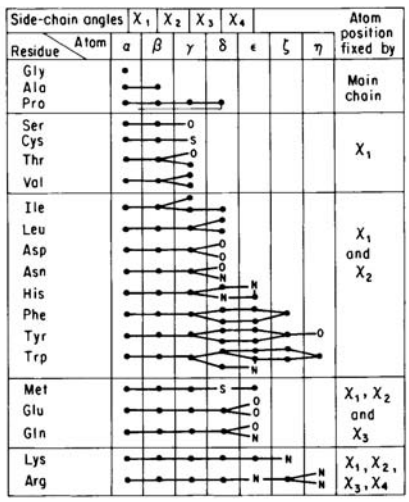
In the most general case, this is not a problem we can solve. If the mutated side chain causes a tremendous rearrangement of the protein, the new structure will not be predictable using current methods. Fortunately, this is rarely the case. Rather, we need to worry about residues near the mutation. Typically, we can assume that the backbone changes little. So our main challenge is to figure out how a set of side chains near the mutation repack (see Figure).



We will make a few modeling assumptions to make the problem easier. First, rather than trying to solve this problem by determining the exact coordinates of each atom, we'll make the approximation and assume that bond lengths are fixed and model using rotations around the side chain bonds. This vastly reduces the space of all possible conformations. We will then use the Metropolis algorithm to find a low energy combination for the side chains.

3. Rotamers

In addition to fixing the bond lengths, we can further simplify the problem by treating bond rotations as discrete rather than continuous. Ponder and Richards (1987) were the first to make the observation that the set of side-chain angles in protein structures tend to cluster, and this observation has held up now that many more structures have been solved.

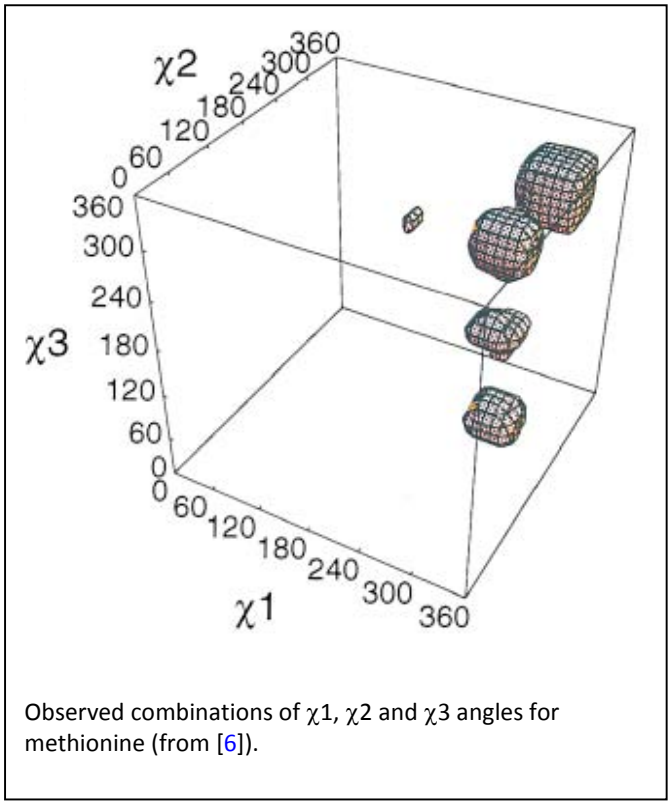


Side-chain angles	χ_1	χ_2	χ_3	χ_4				Atom position fixed by
Residue \ Atom	α	β	γ	δ	ϵ	ζ	η	
Gly, Ala, Pro								Main chain
Ser, Cys, Thr, Val								χ_1
Ile, Leu, Asp, Asn, His, Phe, Tyr, Trp								χ_1 and χ_2
Met, Glu, Gln								χ_1, χ_2 and χ_3
Lys, Arg								$\chi_1, \chi_2, \chi_3, \chi_4$

Figure 1. Flexibility of amino acid side-chains. The Figure shows the chi angle values required to fix the positions of side-chain atoms in each residue type.

Rotatable bonds in side chains (from [3])

Source: Ponder, Jay W., and Frederic M. Richards. "Tertiary Templates for Proteins: Use of Packing Criteria in the Enumeration of Allowed Sequences for Different Structural Classes." *Journal of Molecular Biology* 193, no. 4 (1987): 775-91.



Observed combinations of $\chi 1$, $\chi 2$ and $\chi 3$ angles for methionine (from [6]).

Source: Kuszewski, John, Angela M. Gronenborn, et al. "Improvements and Extensions in the Conformational Database Potential for the Refinement of NMR and X-ray Structures of Proteins and Nucleic Acids." *Journal of Magnetic Resonance* 125, no. 1 (1997): 171-7.

In the observed protein structures the angles can deviate from these ideal positions, but we can still think about the choices as discrete rather than continuous. We often call the different states that differ only by the rotation of a bond as **rotamers,** short for **rotational isomer**.

Using rotamers allows us to simplify the repacking problem from a continuous optimization problem to a combinatorial one: Which combination of rotamers has the lowest energy. Nevertheless, the problem is still extremely complex. If I consider 20 residues around the site of the mutation and, on average, each residue can occupy one of ten different rotamers there are 10^{20} states to consider, and in a realistic situation we probably want to sample more states for more residues.

4. Combinatorial Optimization

The problem with any attempt to optimize a function with hundreds variables is that you are very likely to find only a local energy minimum. But there are algorithms that will help us explore complicated surfaces and find "non-local" minima. These algorithms still don't guarantee they will get the global minimum, but they have a larger radius of convergence. The basic ideas behind the algorithms are very general, and you will see them in lots of other contexts.

To understand it we need to introduce two concepts: 1. Monte Carlo techniques and 2. Metropolis sampling.

Those of you who took 6.00 will be familiar with Monte Carlo search. This technique allows us to compute a complicated function by randomly sampling the underlying variables. In 6.00 you used it to compute lots of things, including the value of pi and games of chance.

In 6.00 you studied "guess and check" algorithms. In these algorithms, you generate random values of the parameters and test if they are good. A guess-and-check energy optimization algorithm in 1D would randomly sample x-coordinates, compute the energy and treat the lowest value as the minimum. This is a type of Monte Carlo algorithm. It relies on random selection of a state. These algorithms are named after the city in Monaco that is famous for gambling casinos.

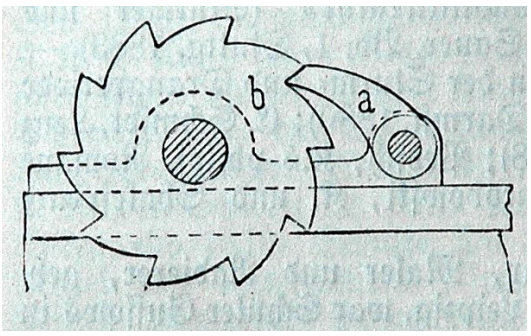
This guess-and-check approach can only work if a random search has some hope of finding an optimal or near-optimal solution. In 1D problem, depending on how close you need to get to the minimum, you might be OK. But in our real-setting, the combinatorial complexity is too much for such an approach. Random search through the 10^{20} combinations of rotamers in our example is unlikely to be productive.

A common solution to reduce the size of the search space derives from physical principles. The Boltzmann distribution tells us that molecules spend most of their time in low energy states. The probability of being in state A is given by

$P(A) = \frac{e^{-E_A/kT}}{Z(T)}$ and the relative probability of two states is given by

$\frac{P(A)}{P(B)} = \frac{e^{-E_A/kT}}{e^{-E_B/kT}} = e^{(E_B - E_A)/kT}$.

How do molecules find the lowest energy states without any knowledge of the global energy landscape? The key turns out to be the frequency at which local transitions occur. Imagine a system, be it an atom or a protein, in some particular state on our 1D model. It makes a random state transition (for example, a side chain gets knocked into a new position due to thermal fluctuations). If it goes to a lower energy state, it tends to stay there because it lacks the energy to move uphill. Similarly, if it goes uphill, it isn't likely to stay there, because it can get knocked back down into the lower energy state. There are also some states that it is very unlikely to be able to move to at all because the energy of these states is much higher than kT.



The behavior of molecules can be thought of as analogous to a ratchet mechanism. They bias movement to lower energy states, but they preserve the ability to jump some barriers – those with energy differences that are small with respect to kT.

Metropolis Algorithm

The **metropolis algorithm** uses the principles of the Boltzmann distribution to sample states in a way that helps it find minima. Let's start with a system in some particular state. It could be a particle on A 1D surface or a protein.

Iterate for a fixed number of cycles or until convergence:

1. Start with a system in state S_n with energy E_n
2. Choose a neighboring state at random; we will call it the proposed state : S_{test} with energy E_{test}
3. If $E_{test} < E_n$: $S_{n+1}=S_{test}$
4. Else set $S_{n+1}= S_{test}$ with probability $P = e^{-(Etest-E_n)/kT}$; otherwise $S_{n+1}= S_n$

The last step is equivalent to saying that we choose between the two states using the odds ratio: $\frac{P(Stest)}{P(Sn)} = \frac{e^{-Etest/kT}}{Z(T)} / \frac{e^{-En/kT}}{Z(T)} = e^{-(Etest-E_n)/kT}$, which exactly what would occur in a physical process.

There are a few issues that we need to clarify. You may have been wondering about the parameter here called T. It is the equivalent of temperature in the physical process. Try to simulate what happens in this algorithm when T is very small and when it is large.

What happens when T is large? Say $kT >> E_{test}-E_n$?

$P(S_{test})=e^{0}=1$ and we accept any move. This allows us to search the whole space, but we are not very likely to rest in a minimum. What happens when T is very small? Say $kT<<(E_{test}-E_n)$? Then, $P(S_{test})=e^{-\infty}=0$. We almost never go up-hill.

The T parameter is critical to the algorithm, and various ideas have been proposed about how to set it. In some cases, you can leave T fixed at a low value and get a reasonable search of the energy surface. However, for very complicated energy surfaces we often used something called simulated annealing. The idea is to use an initial high temperature so you can explore lots of valleys and then slowly lower the temperature so you settle in a good minimum. Precisely how you do this is an art, not a science. (The name "annealing" is derived from metallurgy, where heating and cooling cycles are used to reduce the number of defects in a metal and increase the size of the crystalline blocks.)

Another important issue: In our algorithm we spoke of choosing a neighboring state. This is a poorly defined term. In the 1D case, of course I mean a small step to one side or the other. Consider how we would choose a neighboring state if we need to repack a number of side chains. Finally, any implementation of this algorithm needs to define when the algorithm should stop (convergence criteria).

How do we make probabilistic choices in a computer program?

In order to implement the metropolis algorithm, we need to be able to choose states randomly. We rely on the ability of the computer to produce what is known as a pseudo-random number. Almost every programming language has the ability to produce a series of numbers between zero and one that look like they are chosen completely at random. These numbers have two important properties: (1) it is very hard to predict the next number in a sequence and (2) the numbers are uniformly distributed from zero to one.

If you consider the histogram of these pseudo-random numbers, you will realize that $P(rand<X)=X$ for $0<X<=1$

So to implement line 4 of the algorithm, we can write

if rand()<= $e^{-(Etest-E_n)/kT}$:$S_{n+1}= S_{test}$

The approach we have just described allows us to find a very approximate solution. In particular, we have not allowed the backbone to change to accommodate the new side

chains. We have also restricted the rotamers to discrete choices, even though we know that the true values are continuous. Two approaches will allow us to refine these approximate structures to more accurate ones:

1. Energy minimization
2. Molecular dynamics

Energy Minimization

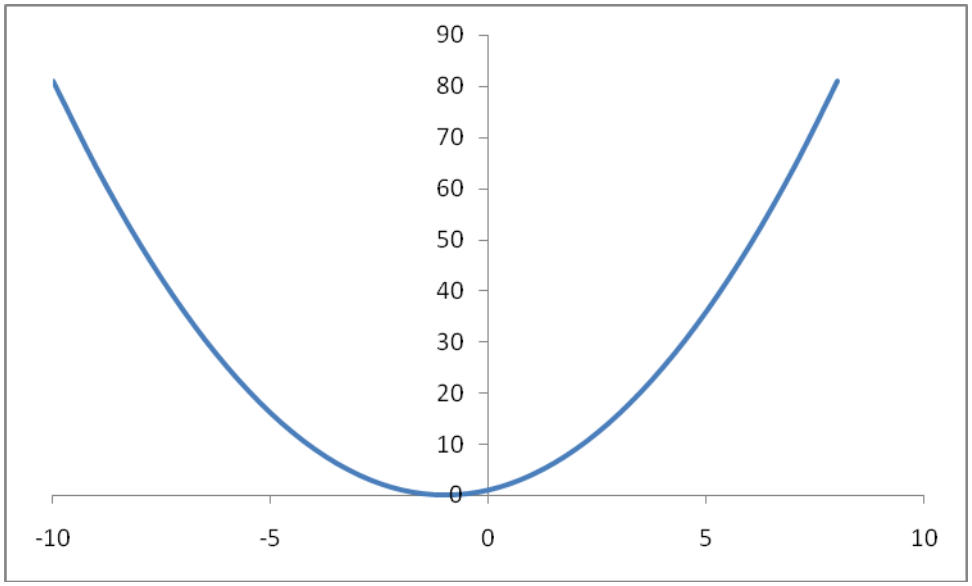
One approach to trying to fix the approximate structures derived from modeling would be to think of them like an elastic object that has been deformed. It will try to relax back to its equilibrium state. So perhaps simulating the physical forces will fix our models. We can attempt to use the potential energy equations described earlier in these notes.

Once we have potential energy equations and a structure, how do we find a new structure with lower energy? One of the most fundamental ways is by energy minimization. You can think of this in direct analogy to minimizing a simple one-dimensional function, except that here we have hundreds of coupled equations.

How would you go about finding a minimum of f(x), a function of one variable?

Look for regions where f'(x)=0; these are "critical points"; then use second derivative to determine if it's a maximum or minimum. For example:

Consider f(x)=x^2+2x+1, a parabola with minimum at -1



f'(x)=2x+2=0

so x=-1

The situation is analogous for functions of many variables:

Look for places where the gradient grad(f) = 0

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n} \right).$$

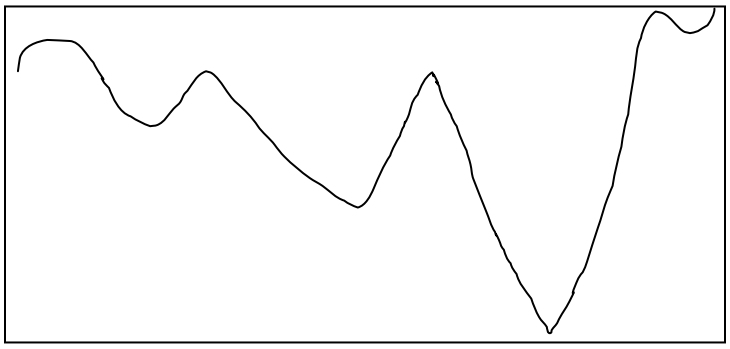
Then use Hessian matrix of second order partial derivatives to determine if the point is a minimum, maximum, etc.

We'd like to take a big equation for the energy of the protein and solve for its minimum.

This isn't practical to do analytically in very high dimensions.

However, there are lots of alternatives to the analytical approach.

Here is a simple method for finding minima known as **gradient descent**. Let's imagine we have some complicated energy function. I'll draw a function of one variable. You might be able to picture a complicated function of two variables. Our system has hundreds of variables.



Here is how gradient descent works: We start with some point x_0 and compute the gradient.

The negative of the gradient is pointing in the direction that decreases most rapidly.

So we can compute a new position

$$\vec{x}_1 = \vec{x}_0 - \varepsilon \nabla U_0(\vec{x}_0)$$

where epsilon is a small number that determines the size of the step.

We can keep iterating this.

When will the solution become stable?

With most functions, this will converge on a solution fairly rapidly. In those cases we are guaranteed to find a minimum, but we don't know if it will be a global or local minimum.

For protein structures we can start with our current structure based on the homology model. That will be on one of these hills. Then we can compute for every atom the best direction in which to move it and iterate this process as we go down the hill.

Molecular dynamics

The atomic forces can be used for more than just minimization. In principle, if we have a sufficiently accurate description of the forces, we should be able to **simulate** what actually is going on in solution. This approach is called **molecular dynamics**:

Once you have the forces, you can write down these equations:

given the position $x_i(t_0)$ and velocity $v_i(t_0)$ of the ith particle at the starting time t_0, the position and velocity of the particle a short time later, at t_1, are given by

$$x_i(t_1) = x_i(t_0) + v_i(t_0) \times (t_1 - t_0) \quad (1)$$

$$v_i(t_1) = v_i(t_0) + \frac{dv_i(t_0)}{dt}|_{t_0} \times (t_1 - t_0) \quad (2)$$

Equation (2) becomes the following from Newton's equation of motion,

$$v_i(t_1) = v_i(t_0) + \frac{F_i(t_0)}{m_i} \times (t_1 - t_0)$$

This relatively simple equation allows us to predict the motion of an atom in time. Thus, we could in principle simulate anything from a minor relaxation of a deformed protein to an entire folding process. However, there are a lot of details to work out here.

1. The most important point is the motion of atom *i* is a function of the positions of all the other atoms, as all of these contribute to $F_i(t)$. So we have a horribly complicated set of equations of motion that will need to be solved numerically.

2. In addition to the terms we have already discussed, the parameters for bond lengths and angles need to be incorporated in the force calculations.

3. You need to choose initial values for the position and velocity.

All these problems have been solved, and molecular dynamics simulations (MD) are a very powerful tool for sampling the range of physical motion that can take place. They were crucial calculations in the early days of protein structure because they helped convince people that proteins are not rigid. The techniques for determining protein structure give static pictures. MD animates these pictures, giving the correct impression that the proteins are jiggling around. These calculations have been used to model many interesting questions, such as how oxygen gets in and out of the heme pocket in myoglobin.

Unfortunately, MD is limited by something known as the radius of convergence. We can only simulate relatively short time scales.

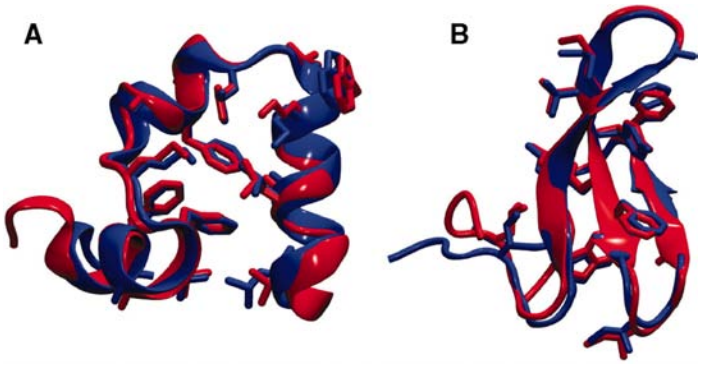


Figure from Shaw *et al.* (2010). Fig. 1 Folding proteins at x-ray resolution, showing comparison of x-ray structures (blue) (*15*, *24*) and last frame of MD simulation (red): (**A**) simulation of villin at 300 K, (**B**) simulation of FiP35 at 337 K. Simulations were initiated from completely extended structures. Villin and FiP35 folded to their native states after 68 μs and 38 μs, respectively, and simulations were continued for an additional 20 μs after the folding event to verify the stability of the native fold.

Until recently, heroic efforts using supercomputers have been only able to simulate folding of a tiny protein (36 amino acids) on a time scale of one micro second, resulting in a structure with an RMSD of 4.5 angstroms (Duan and Kollman (1998) Science 282:740; see Schaeffer, *et al.* (2008) Current Opinion Struct. Biol. 18:4 for more recent references). However, Shaw, *et al.* (2010) Science 330:341 have made a dramatic advance in this area. Using a specially built computer they have been able to simulate one millisecond of dynamics for a small protein. This was long enough to observe multiple folding and unfolding events. For two proteins, they simulated folding from an extended conformation to the correct final folded form. In 2011, they were able to simulate the folding and unfolding of twelve small proteins, the largest of which was 80 amino-acids [8].

Alternative uses of molecular dynamics

Even if computing speeds increase dramatically, simulating the time evolution of a single protein conformation will be of limited utility in understanding the behavior of a bulk solution. Most often, we want to understand the behavior of the ensemble of molecules. Consider the protein folding problem. Since the unfolded state is an ensemble, the amount we can learn from a single simulation is inherently limited.

There have been a number of interesting developments in protein structure prediction and folding simulation that, in one way or another, attempt to sample from the overall distribution. One conceptually simple and very successful approach was pioneered by the Pande lab and is called Folding@home. They distribute folding simulations to the computers of tens of thousands of volunteers where the jobs run when the computer would otherwise be idle. This is still a very small ensemble compared to even a very dilute solution, but it has one clear advantage. If we assume that folding follows first-order kinetics, then the probability of observing a folding event is linear in the number of molecules we sample and in the length of time we simulate. So, distributing a job to 10,000 computers is equivalent to a 10,000-fold increase in the time-scale over which we can search for a folding event. Even with Moore's law, it takes a long time to get a 10^4 increase in computing speed.

How do you measure the distance between two protein conformations?

First, we have to agree that the actual coordinates of a protein are not what we want to measure. I can rotate or translate all of these in a uniform way and I haven't changed the energy or the structure.

So if I have two structures that are identical, except for a translation or a rotation, it's not interesting.

To determine the similarity of two structures, we attempt to superimpose them.

We search for the rigid body rotation and translation that minimizes this quantity:

$$\mathrm{RMSD}(\mathbf{v}, \mathbf{w}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \|v_i - w_i\|^2}$$

where v and w are the coordinates for pairs of equivalent atoms (usually *Cα* and sometimes *C*,*N*,*O*,*Cβ*).

An RMSD of **0.5 Å** is very good. It has been estimated that the alpha carbons in independent determinations of the ***same*** **protein** can have this level of variation [9]. The useful range for a model would be less than 3 angstroms.

Even at 3 angstroms, you can't refine the model just by running a simulation of the atomic forces.

The key in molecular modeling is finding ways to move large distances at low computational cost. We will return to this topic in future lectures.

Molecular Design Part 2 Drug Design

In the previous lecture we examined ways in which we could modify the specificity of a protein. In today's lecture we will look at ways to identify small molecules that can bind exisiting proteins, a topic which is obviously of great interest to the pharmaceutical industry.

How to design small molecule ligands

The search for small molecule pharmaceuticals can be divided into two types:

Analog-based design

Structure-based design

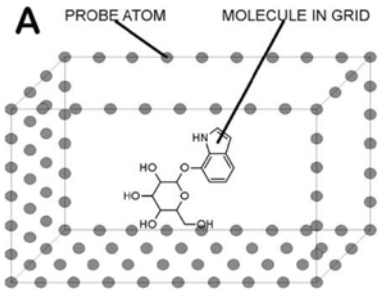
Analog-based design is useful one doesn't know what the target is, but there is a compound with some activity in an assay. Many analogs (variants) of the lead compound are made and tested for efficacy. Aspirin is good example of analog-based design. This drug has many effects, including reducing fever, reducing pain and preventing stroke. The first leads were identified in ancient times when willow bark was used as a therapeutic for fever, a fact known to Hippocartes. Salicylic acid, the active components of willow bark, was identified in the mid nineteenth century, but it was very bitter and caused digestive problems. In 1897, chemists at Bayer determined that acetylsalicyclic acid, an **analog** of salicylic acid, retained the therapeutic properties but was easier to ingest. Yet, it was only in 1971 that the first target was identified (which led to a Nobel prize). Clearly, it is possible to produce safe an effective drugs by testing variants of a lead compound without understand the mechanism by which the compounds function.

One of the principal approaches to modern analog-based design is known as **QSAR** (pronounced Quasar), which stands for quantitative structure-activity relationship. Analog-based design is a collection of data-mining techniques that look for new chemical entities that have similar properties to a lead compound. One tests a number of variants of the lead compound and then looks for a correlation between the free energy of binding and particular structural properties of the variants such as size, hydrophobicity, etc. This allows future experiments to focus on particular types of analogs. Of course, there is no guarantee that such a relationship exists. Nor can we be sure that if one is found it will apply to untested compounds.

One important challenge for this approach is to identify properties that are likely to be physically meaningful. Early analyses in the 19th century by Meyer and Overton (cited in [4]) found relationships between overall hydrophobicity and the effectiveness of anesthetics, but such trends are not likely to be of help in discovering compounds with high affinity and activity for particular molecular targets.

A number of types of descriptors are in use. A very common one is called Comparative Molecular Field Analysis (CoMFA). The molecules are aligned and placed in a grid. Imaginary probe atoms are then tested at each point on the grid to determine their energy

of interaction with the molecule. This gives a three-dimensional map of the hydrophobic, steric and electrostatic properties that are common to the ligands.



Method of computing 3D properties for QSAR (from [4]).

Structure-based design begins with the structure of protein and looks for a small molecule (<500 Da) that will bind tightly. This approach has important differences from the engineering of protein-protein that we examined previously. In a case like the redesign of the GCSF-GCSFR interface, the **wild-type hormone formed a great scaffold**. If our changes were conservative, we could assume that the overall interface and the hormone structure remained the same.

For small molecules:

1. We don't know where the ligand is going to bind on the surface of the protein.
2. We don't just have 20 amino acids to choose from, but millions of possible small molecules.
3. Depending on the potential ligand, we may have more degrees of freedom in the ligand.

Structure-based design

There has been considerable success in using the three dimensional structures of protein-ligand complexes to improve on the specificity and affinity of small molecules that were identified already. A survey of the success stories can be found here [10] and includes design of compounds to inhibit HIV protease and the development of Tamiflu, which was widely used the H1N1 flu pandemic. In these cases, the structures are analyzed to determine what changes could be made to the ligand to increase its affinity for the target.

There is also tremendous effort to use these methods to discover molecules that will bind to a protein of interest without needing an initial lead. In this general approach there are two primary problems that must be solved:

1. Docking

2. Scoring

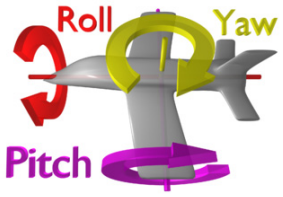
Docking:

In the "docking" phase, algorithms attempt to predict the position and orientation of a ligand on a target. (This is also referred to as the "**pose**" of the ligand). In these calculations, it is typical to ignore solvent and assume that the protein is rigid. With those assumptions, there are six variables we need to consider: translation in three perpendicular axes combined with rotation about three perpendicular axes (yaw, pitch, roll).

Although most docking programs rely on a rigid protein, they allow the ligand to change conformation. The optimization problem is typically solved by one of three methods:

- **Systematic search**
- **Stochastic search**
- **Molecular dynamics simulations**

Systematic search corresponds to the "guess and check" algorithms of 6.00. Stochastic search includes the metropolis algorithm and related approaches. Molecular dynamics simulations were covered in the last lecture.



Scoring functions. In order to determine which ligand "pose" is best, we need a way to score the interactions. There are three methods that are the most common:

- Force field-based score
- Knowledge-based scoring
- Empirical scoring

The force field based scores are what we have looked at so far. Their strength is that they derived from sound physical principles. However, we have seen that in order for them to produce accurate results, especially with regard to solvation, they need extremely fast computers.

Knowledge-based scores derive the frequency of certain types of interactions and conformations from databases. The following equations for a knowledge based potential come from reference [11]. Let r be the distance between two atoms, and $\rho_{ij}(r)$ be the number of atoms pairs of types i and j that are separated by distance r in the database, and $\rho^{*}_{ij}(r)$ be the number of such pairs in a reference state (see the original article for a definition of the reference state). If we assume that the ratio of these numbers will be

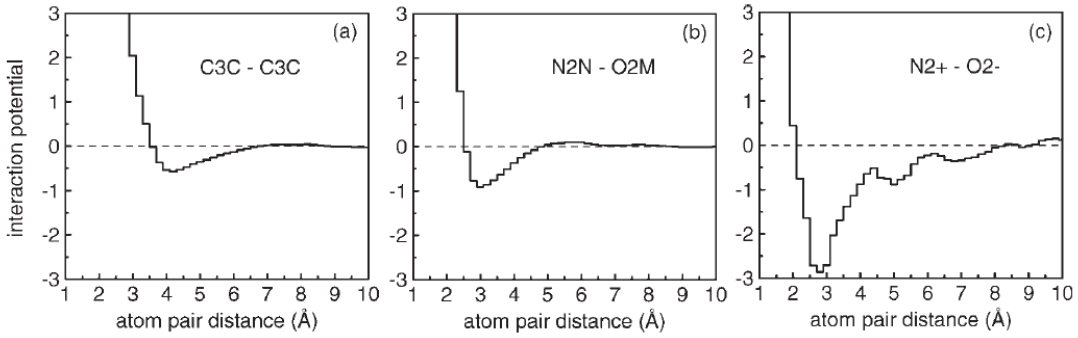
distributed according to the Boltzmann distribution, then we can derive an empirical potential $u_{ij}(r)$ from ratio of the frequency of the observations:

$$u_{ij}(r) = -k_B T \ln[g_{ij}(r)], \quad g_{ij}(r) = \rho_{ij}(r)/\rho^*_{ij}(r)$$

Various atom types are defined for these functions, such as "aliphatic carbon" and "aromatic carbons" to capture the complex chemistry.

Below are some sample potentials for "aliphatic carbons bonded to carbons or hydrogens only (C3C)", "Amide nitrogens with one hydrogen (N2N), "oxygens in carbonyl groups of the mainchain (O2M)", "Guanidine nitrogens with two hydrogens (N2+)" and "Oxygens in carboxyl groups (O2-)".

One challenge for knowledge-based approaches is that there are likely to be a number of different types of binding sites for ligands. For example, polar binding pockets should have different distributions of atoms than non-polar pockets. In principle, this problem could be overcome by splitting the training data into categories and developing separate potentials for each type of pocket.



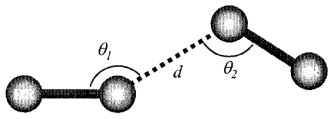
Empirical scoring methods have a similar functional form to force field-based scores, but include additional terms that capture missing energetic components. For example, the program x-score [1] breaks the binding energy into the following components:

$$\Delta G_{\text{bind}} = \Delta G_{\text{vdw}} + \Delta G_{\text{H - bond}} + \Delta G_{\text{deformation}} + \Delta G_{\text{hydrophobic}} + \Delta G_0$$

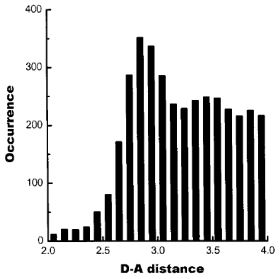
Each term has a set of parameters that are derived from **structures for which the Kd is known**. The vdW term looks a lot like our force-field term, but all the hydrogen atoms are ignored. $\Delta G_{\text{deformation}}$ tries to capture the entropic effects of binding. One way to do this is to assign a penalty term for every bond that can rotate in the free ligand but is fixed in the bound ligand. Similarly, the hydrophobic term tries to capture the cost/benefit of burying varies types of atoms. In these cases, the functional form is arbitrary. You could choose almost any sort of equation and then try to fit the free parameters to experimental

data. The final term is derived from the regression analysis and sweeps up remaining free energy changes such as loss of translational and rotational freedom.

The hydrogen bond term from one of these potentials is shown below. First, the geometry of the H-bond is defined in terms of two angles between the heavy (non-hydrogen) atoms, since the hydrogens are not seen in the crystal structure.



Then the H-bond term is set to $\mathrm{HB}_{ij} = f\left(d_{ij}\right) f\left(\theta_{1,ij}\right) f\left(\theta_{2,ij}\right)$, where each of the functions is obtained by looking at the frequency of particular angles and distances in the databases.



Observed distribution of distances between donor and acceptors

The specific parameters in each term are determined by fitting the equations to experimental data for protein-ligand interactions. The fitted data from one of these methods are shown below.

Do these models actually work?

Many of these algorithms have been successful in building accurate models of the ligand bound to its target. However, they are not very good at distinguishing true ligands from decoys. So you need to be thoughtful in how to apply these algorithms. The will help enrich a pool of ligands for real binders, and can be very useful in making sure the experimental resources are used wisely. As a result, these algorithms are often used for **de-selection**, which is the process of predicting really bad compounds to be filtered out of the experimental process.

Which proteins make good drug targets?

Genomic and proteomic technologies have dramatically sped up the earliest steps of the drug discovery by helping to identify protein targets that are believed to be important in a disease. With so many potential targets, it has become increasingly important to figure out which of these targets can be most effectively targeted by a drug. A consensus is emerging in the pharmaceutical industry that some proteins are much harder, perhaps even impossible, to target than others, and there is great interest in computational methods that could reveal which proteins are the most "druggable." Examples of proteins that are

17

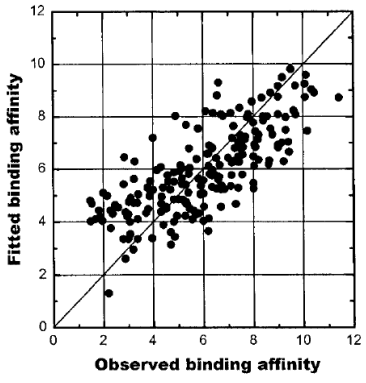


Figure 4. Correlation between the observed binding affinities of the 200 protein-ligand complexes in the training set and the fitted values given by X-CSCORE (in pK_d units).

From reference [1]

Source: Gao, Ying, Renxiao Wang, et al. "Structure-based Method for Analyzing Protein–Protein Interfaces." *Journal of Molecular Modeling* 10, no. 1 (2004): 44-54.

believed to be undruggable include interleukin-1 beta-converting enzyme 1 (for anti-inflammatory therapy), phosphotyrosine phosphatase 1B (diabetes), cathepsin K (arthritis and osteoporosis) and HIV integrase.

A number of approaches have been proposed for identifying druggable targets (reviewed in Hajduk et al. 2005). One of the simplest approaches is to assume that if a protein has a lot of sequence homology with the target of a known drug, then it too is likely to be druggable. Thus, there is a belief in the pharmaceutical industry that G-protein coupled receptors, ion channels, kinases and proteases make good drug targets. However, this approach is inherently self-limiting.

An alternative approach analyzes the three dimensional protein structures to determine how druggable a particular protein is (rather than a whole family). The challenge here is that there is no experimental or computation way to test a single protein against all possible drugs. It has been estimated that the number of potential drugs is on the order of 10^{60}.

Cheng et al. (2007) proposed a method to predict the "maximum achievable binding energy" – the maximum possible interaction energy of a protein with **any** ligand. If this value is low for a protein, then there is no point in searching for ligands, because they will not make good drugs. They argue that solvation effects will dominate the calculation of maximum achievable binding energy. They then point out that based on concepts we will examine later (see Lipinski's rules), most orally available drugs will have the same size and charge. As a result, the value of energy terms from van der Waals, electrostatics and loss of rotational and translational energy will be roughly the same for the binding of any protein to its optimized orally available drug. Similarly, the cost of desolvating the ligand will also be approximately constant. They propose that the dominant force in the calculation of the maximum achievable binding energy will be the energetic cost of desolvating the ligand binding site on the protein, which will depend on the curvature of the binding site and the fraction of the site that is hydrophobic. Somewhat surprisingly, these two features seem sufficient to distinguish druggable and undruggable proteins in their dataset. Perot et al. [12] survey a number of other methods for detecting druggable proteins.

A number of studies have analyzed the structures of proteins bound to other proteins, non-protein ligands and drugs [13]. As you might expect, there are big differences between protein-protein and protein-small molecule interfaces. Protein-protein interfaces tend to be composed of many small binding pockets, whereas ligands bind to fewer, but but bigger pockets. In cases where there are structures available for both the bound and unbound form, it is clear that ligands cause an increase in the size of the pockets. Thus, it may be necessary to consider more than just the static structure of an unbound protein in order to know if it is druggable. One approach would be to use molecular dynamics or other techniques to determine if there are alternative low-energy conformations with larger pockets. These might be conformations to which a ligand could bind and stabilize.

ADME/T

Broad considerations for drug design

A crucial question in pharmacology is what happens after a drug is administered. The main concerns are referred to as **ADME/T**: **Absorbtion, distribution, metabolism, excretion and toxicity.** The first four of these (ADME) are often grouped together under pharmacokinetics.

Absorption refers to how the compound gets taken up from the site of administration (oral, topical, injected). If it is swallowed, it has to get through the digestive tract into the blood stream, for example.

Distribution: how does it get to the target organ.

Metabolism: how does it get broken down? Is the administered compound the active one, or is it one of the metabolites? Are the metabolites toxic?

Excretion/Elimination: what happens to the compounds and their breakdown products. If they are not eliminated at all, there will be toxicity. If they are eliminated too fast, they are not active enough. Are they toxic at the site of excretion?

The path a typical drug takes is illustrated in the figure below.

Error!

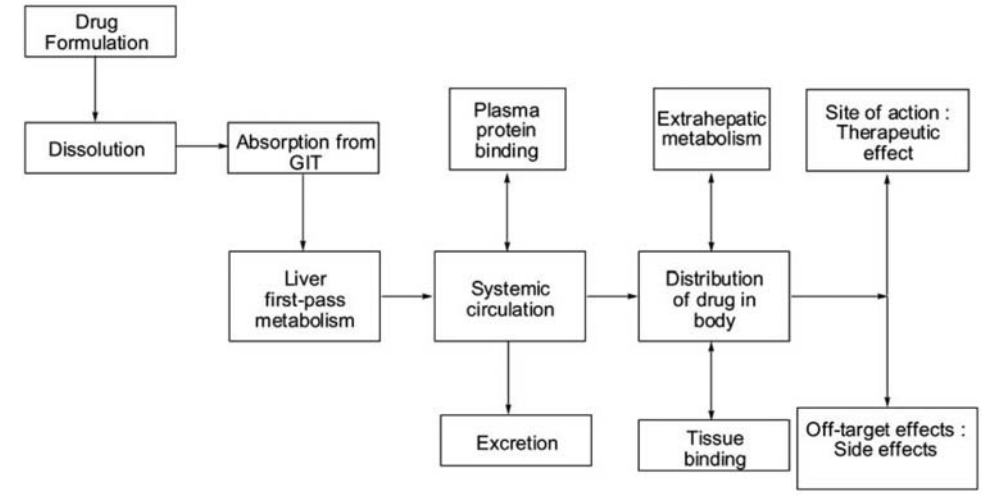


Fig. (1). Processes involved in the 'random walk' of an orally administered drug from its site of administration to its site of action (modified from [7]).

Figure from Kharkar (2010) [7].

Proteins tend to make relatively poor drugs because they typically must be injected rather than ingested and are metabolized quickly.

Most drugs tend to be small molecules. In fact, almost all have particular properties that can be summarized by a few simple rules.

Lipinski's Rule of Five states that compounds with two or more of the following characteristics are likely to make poor drugs because of bad oral absorption:

- More than 5 H-bond donors
- Molecular weight >500
- lipophilcity is high (log P>5, where P is the partition coefficient between Octanol and water)
- Sum of N's and O's (a rough measure of H-bond acceptors) > 10

Others have included the following warning signs:

- Polar surface area > 140 $Å^2$
- More than one formal charge

These rules, more appropriately described as guidelines, do not cover drugs that are derived from natural products, for which other absorption mechanisms are involved, or injected drugs. However, it is clear that, in general, that small molecules rather than proteins will make better drugs.

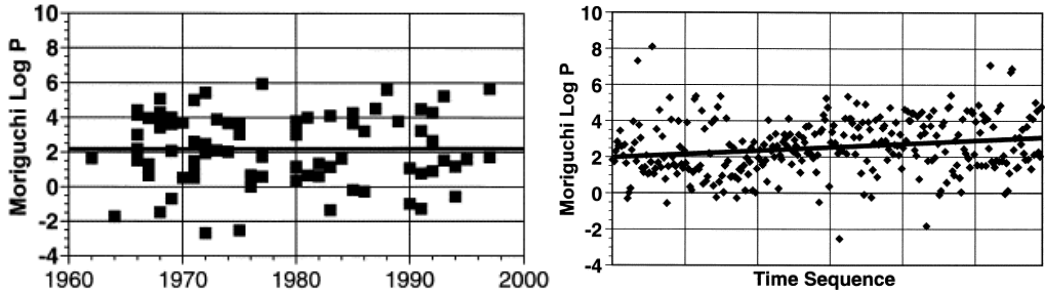
It is important to realize that because the number of potential compounds is huge, most have never been experimentally tested for logP, and this value must be estimated. Because the compound will encounter a variety of environments in the stomach, gut and in the blood, which vary in pH from 1 to 8, there is no single value for logP that will reflect all these conditions.

The requirements for low lipophilicity and also a limited number of polar atoms reflects the need for a balance between hydrophobicity on the one hand, which allows the molecule to get across membranes in the intestine and the blood-brain barrier, and solubility in the gut and blood. Lipinski has since created a number of other rules, cited in [7], that help predict compounds that are likely to work in the central nervous system, lungs, etc.

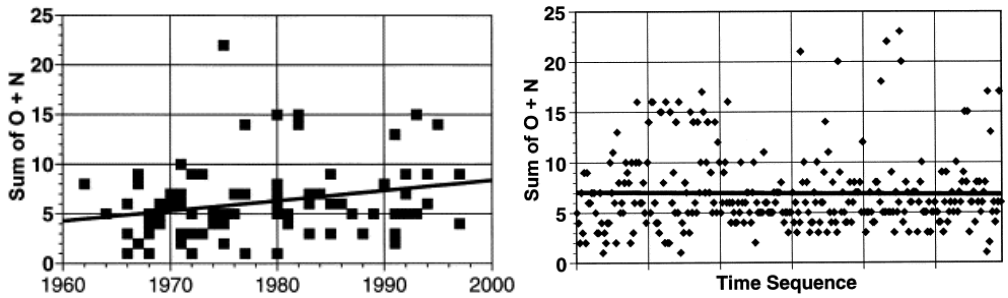
Lipinski also compared lead compounds for Merck and Pfizer and discovered that the method used for drug discovery biased the types of compounds that were discovered. Merck tends to focus on "rational design" of the type described in these notes, while Pfizer focused on high-throughput screens. Compounds from both companies showed an increase in molecular weight with time. The Merck compounds show no increase in lipophilicity, but do show an increase in H-bonds. The opposite is seen for the Pfizer compounds. Each screening process tends to select for properties that will increase

affinity in the assay (computational free energy for Merck and in vitro binding for Pfizer), at the expense of bioavailability.

Error!



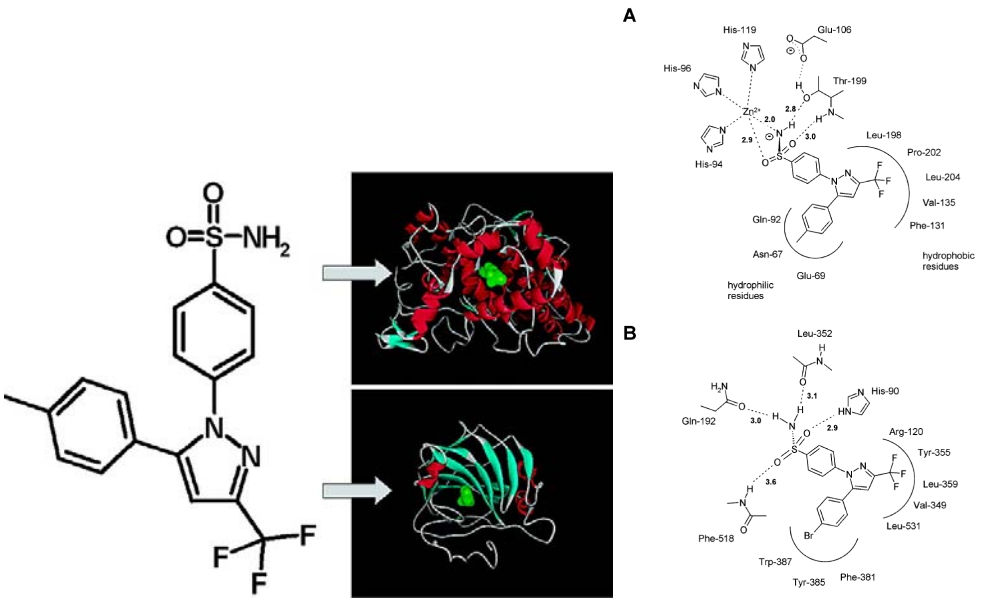
Top: Merck compounds (left) show now increase in logP over time, while Pfizer compounds do. Bottom: By contrast, the Merck compounds (left) show an increase in H-bonds not seen in the Pfizer compounds (right). Figures from [5].



Source: Lipinski, Christopher A. "Drug-like Properties and the Causes of Poor Solubility and Poor Permeability." *Journal of Pharmacological and Toxicological Methods* 44, no. 1 (2000): 235-49.

There are many efforts underway to predict other aspects of ADME/T. These focus on particular steps in the process, such as oral bioavailability, binding to plasma proteins, hepatic metabolism and specific mechanisms of toxicity. (Reviewed in [7]).

Some off-pathway effects (whether beneficial or detrimental) are likely to be caused by a single molecule binding two different proteins. There are not enough data yet to know whether this occurs because the binding sites on the protein are similar or because of two very different modes of interaction. However, there is a least one case with that shows two proteins of completely different structure binding a ligand with similar pockets (see figure). Algorithms that can detect such similar pockets could be very useful in predicting toxicity or alternative uses of a drug. COX-2 inhibitors have been shown to bind carbonic anhydrase, in addition to the expected COX-2 proteins, and have be proposed as potential treatments for glaucoma, which is treated by inhibition of carbonic anhydrase.



The COX-2 specific drugs celecoxib and valdecoxib also bind carbonic anhydrase with nanomolar affinity despite the lack of any structural homology (left). However, the binding pockets have some similarity in shape and chemical properties (right). See [2] for details.

Source: Weber, Alexander, Angela Casini, et al. "Unexpected Nanomolar Inhibition of Carbonic Anhydrase by COX-2-selective Celecoxib: New Pharmacological Opportunities due to Related Binding Site Recognition." *Journal of Medicinal Chemistry* 47, no. 3 (2004): 550-7.

References:

1. Gao, Y., R. Wang, and L. Lai, *Structure-based method for analyzing protein-protein interfaces.* J Mol Model, 2004. **10**(1): p. 44-54.
2. Weber, A., et al., *Unexpected nanomolar inhibition of carbonic anhydrase by COX-2-selective celecoxib: new pharmacological opportunities due to related binding site recognition.* J Med Chem, 2004. **47**(3): p. 550-7.
3. Ponder, J.W. and F.M. Richards, *Tertiary templates for proteins. Use of packing criteria in the enumeration of allowed sequences for different structural classes.* Journal of molecular biology, 1987. **193**(4): p. 775-91.
4. Sarkar, A. and G.E. Kellogg, *Hydrophobicity--shake flasks, protein folding and drug discovery.* Curr Top Med Chem, 2010. **10**(1): p. 67-83.
5. Lipinski, C.A., *Drug-like properties and the causes of poor solubility and poor permeability.* J Pharmacol Toxicol Methods, 2000. **44**(1): p. 235-49.
6. Kuszewski, J., A.M. Gronenborn, and G.M. Clore, *Improvements and extensions in the conformational database potential for the refinement of NMR and X-ray structures of proteins and nucleic acids.* Journal of magnetic resonance, 1997. **125**(1): p. 171-7.
7. Kharkar, P.S., *Two-Dimensional (2D) In Silico Models for Absorption, Distribution, Metabolism, Excretion and Toxicity (ADME/T) in Drug Discovery.* Current Topics in Medicinal Chemistry, 2010. **10**(1): p. 116-126.
8. Lindorff-Larsen, K., et al., *How fast-folding proteins fold.* Science, 2011. **334**(6055): p. 517-20.

9. Chothia, C. and A.M. Lesk, *The relation between the divergence of sequence and structure in proteins.* The EMBO journal, 1986. **5**(4): p. 823-6.
10. Talele, T.T., S.A. Khedkar, and A.C. Rigby, *Successful applications of computer aided drug discovery: moving drugs from concept to the clinic.* Curr Top Med Chem, 2010. **10**(1): p. 127-41.
11. Huang, S.Y. and X. Zou, *An iterative knowledge-based scoring function for protein-protein recognition.* Proteins, 2008. **72**(2): p. 557-79.
12. Perot, S., et al., *Druggable pockets and binding site centric chemical space: a paradigm shift in drug discovery.* Drug Discov Today, 2010. **15**(15-16): p. 656-67.
13. Fuller, J.C., N.J. Burgoyne, and R.M. Jackson, *Predicting druggable binding sites at the protein-protein interface.* Drug Discov Today, 2009. **14**(3-4): p. 155-61.
14. Boas, F.E. and P.B. Harbury, *Potential energy functions for protein design.* Curr Opin Struct Biol, 2007. **17**(2): p. 199-204.
15. Cheng, A.C., et al., *Structure-based maximal affinity model predicts small-molecule druggability.* Nat Biotechnol, 2007. **25**(1): p. 71-5.
16. Ekins, S., J. Mestres, and B. Testa, *In silico pharmacology for drug discovery: methods for virtual ligand screening and profiling.* Br J Pharmacol, 2007. **152**(1): p. 9-20.
17. Ekins, S., J. Mestres, and B. Testa, *In silico pharmacology for drug discovery: applications to targets and beyond.* Br J Pharmacol, 2007. **152**(1): p. 21-37.
18. Hajduk, P.J., J.R. Huth, and C. Tse, *Predicting protein druggability.* Drug Discov Today, 2005. **10**(23-24): p. 1675-82.
19. Halperin, I., et al., *Principles of docking: An overview of search algorithms and a guide to scoring functions.* Proteins, 2002. **47**(4): p. 409-43.
20. Huang, N., B.K. Shoichet, and J.J. Irwin, *Benchmarking sets for molecular docking.* J Med Chem, 2006. **49**(23): p. 6789-801.
21. Kitchen, D.B., et al., *Docking and scoring in virtual screening for drug discovery: methods and applications.* Nat Rev Drug Discov, 2004. **3**(11): p. 935-49.
22. Leach, A.R., B.K. Shoichet, and C.E. Peishoff, *Prediction of protein-ligand interactions. Docking and scoring: successes and gaps.* J Med Chem, 2006. **49**(20): p. 5851-5.
23. Sousa, S.F., P.A. Fernandes, and M.J. Ramos, *Protein-ligand docking: current status and future challenges.* Proteins, 2006. **65**(1): p. 15-26.
24. Warren, G.L., et al., *A critical assessment of docking programs and scoring functions.* J Med Chem, 2006. **49**(20): p. 5912-31.

20.320 Analysis of Biomolecular and Cellular Systems
Fall 2012