Binding affinity estimation using the Linear Interaction Energy method (LIE): application to HIVRT

Hugo Gutiérrez de Terán, Jens Carlsson, Johan Åqvist

November 21, 2008

Contents

1 Introduction

In this practical the free energy of binding to HIV-RT will be calculated for two ligands from the NNRTI family. This example is adapted from the paper Carlsson *et al.*, J. Med. Chem. 51:2648-2656 (2008). We will focus on the inhibitors **42** and **62**, as denoted in the original reference. We will make a further evaluation of the docking poses selected from the practical about docking. Note that you can adapt this protocol to estimate the free energy of binding of any docking pose, and easily extend it to evaluate any other inhibitor from the same family.
We will run molecular dynamics (MD) simulations of both the protein-ligand complex and the solvated ligand using the software `Q`. The energies will be extracted and related through the LIE equation to obtain an estimated free energy of binding. We will learn the use of different modules of the `Q` software to elucidate the molecular interactions responsible of ligand binding. The estructures will be also graphically analyzed using the molecular modelling package `PyMOL`.

1.1 Linear Interaction Energy method

The LIE method for the calculation of the *absolute* free energy of binding (ΔG_{bind}) was first proposed by Åqvist *et al* in 1994. It is a semi-empirical method, which is computationally less expensive than the rigorous free energy perturbation (FEP) method.

The estimated energy of binding is calculated as a linear combination of the differences in the average ligand-surrounding interactions (where surrounding is referred to protein and water, when considering the bound state, or just water if considering the free state). Interaction energies are split into electrostatic and van der Waals terms, and weighted by different factors:

$$\Delta G_{\mathrm{binding}} = \alpha(\langle V_{l-s}^{vdw}\rangle_{\mathrm{p}} - \langle V_{l-s}^{vdw}\rangle_{\mathrm{w}}) + \beta(\langle V_{l-s}^{el}\rangle_{\mathrm{p}} - \langle V_{l-s}^{el}\rangle_{\mathrm{w}} + \gamma \quad (1)$$

Where the brackets denote thermal averages sampled during molecular dynamics (MD) of the electrostatic (*el*) and van der Waals (*vdw*) potential interaction energies for the ligand atoms in the protein (p) and water (w) environments.
The main idea of the method is to consider polar and non-polar contributions to the free energy separately.

- **The polar contribution**, $\beta\Delta\langle V_{l-s}^{el}\rangle$: The scaling factor β=0.5 is theoretically derived from electrostatic linear response theory and yields very good agreement with experimental solvation energies for ionic solutes. For uncharged compounds, FEP calculations have shown that lower β values are necessary and can be assigned by a simple scheme depending on the ligand's chemical nature. For the ligands considered here β=0.43 (group: neutral ligand with no hydroxyl groups).

- **The non polar contribution**, $\alpha\Delta\langle V_{l-s}^{vdw}\rangle$: The non-polar contributions to the free energy of binding are assumed to have a linear relationship with the surrounding van der Waals energies. The scaling factor, α, is empirical, but a value of 0.18 has worked well for a large number of systems, including the system considered here.

  An additional constant, γ, may be required to reproduce absolute binding free energies and is dependent on the hydrophobicity of the binding site, *i.e.* specific for a given protein but conserved on any series of ligands studied on that protein. For HIV-RT the optimal value of this constant term is γ=-10.2 kcal/mol.

The goal of MD in the binding calculations is to generate an ensemble of structures and energies that reproduces thermal equilibrium. These ensembles can then be used to calculate the thermodynamic properties of interest, such as $\Delta G_{\mathrm{binding}}$. A few points and recommendations are worth mentioning:

- **Starting coordinates** are taken from xray structures or homology models. We need to generate a *topology file*, which combines the information contained in the initial PDB file (initial positions of the atoms) and the information contained in the force field for each atom.
- The **forcefield** chosen must have parameters for the protein, the solvent and the ligand. In our case we will use OPLS all atom force field, and the ligand parameters have been adapted manually from the original forcefield.
- The system will be modeled with **spherical boundary conditions**, centered on the chemical group of interest (i.e., the ligand). Water molecules are added before the simulation to fill vacant positions and restraints are used to reproduce bulk water density and polarization near the system boundary. Atoms outside the system boundary are conformationally restrained to initial positions. Charges close to the boundary, as well as those outside the solvent sphere, should be neutralized because of the inability to solvate them.
- **Trick**: If the protein is quite big, like HIVRT, it can be useful to shrink the PDB file within a given margin around the solvent sphere. This trick will to allow save time on unnecesary computations, relative to the bonded terms of protein atoms outside of the system of interest (solvated sphere), and anoying text editing of the original PDB file. Note that only non-bonded interactions involving atoms *inside* the system boundary are calculated.
- To reduce the number of pair interactions, several approximations are introduced:
  - For every atom *i* there is a cutoff (keyword **cutoff**) distance for the treatment of its non bonded interactions: every possible pair *i,j* inside the cutoff is periodically tabulated according to a user defined interval of time steps.
  - Beyond the cutoff the electrostatic interactions are approximated through the local reaction field (keyword **lrf**) approximation, in which a fourth order series expansion of the electric field, E, due to all atoms outside the cutoff is calculated. The force acting on an atom *i* due to the electric field is obtained by:

$$\vec{F_{\mathrm{i}}} = \vec{E_{\mathrm{i}}} \cdot q_{\mathrm{i}}$$

  - All van der Waals forces outside the cutoff are ignored.
  - It is important to note than in any free energy calculation the atoms which energy will be calculated (so called Q atoms in the `Q` programs, *i.e.* the ligand) do not have any of these approximations, and they explicitly see *every* other atom within the sphere of simulation. The complete list of those atoms must be specified in a separate file, *e.g.* `lie.fep`.

2 Methods: Prepare and run the MD simulations

We will need to perform two separate MD simulations of the ligand, that we will call the "water simulation" and the "protein simulation". In both cases the ligand will be solvated by a 18Å sphere of waters, and we will collect the relevant ligand-surrounding interaction energies, which will be used as input in the LIE equation to get an estimated value for ΔG_{bind}.
All MD simulations will be done with the modified version of the forcefield OPLS, as implemented in the software `Q`. All we need is a set of files and executables as listed bellow:

a. A PDB file, containing the coordinates of the ligand (water simulation) or the complex, i.e. the protein plus a selected docking pose of ligand (as obtained in the docking exercise from the previous day). See the section "PDB preparation for details"

b. The files that constitute the *forcefield*

   - A library file for each molecule in the complex, with relevant information about the *atom names*, *atom types*, *partial charges* and *bonds* present in the molecule. For this particular case you have the **Qoplsaa.lib** file, with information about all the protein residues plus the water (TIP3 water model), **HIVRT.lib**, with information about the ligands considered. Note that the ligands have been divided into building blocks, that have been considered as independent residues.
   - A parameter file, with all the molecular mechanics parameters needed for simulate this system: *Van der Waals, bond stereching, angle bending, torsion and improper angle* parameters. This file is called **Qoplsaa.prm**

c. The fep file, which stores the information about which atoms correspond to the ligand (the so called "Q atoms") The ligand-surrounding energies that we will analyze will be calculated based on this selection of atoms.

d. The Q executables, which are stored in your path. There are 3 executables that we will use from the Q package: `Qprep5` to *prepare* the topology, `Qdyn5` to *run* the simulation and `Qcalc5` to *analyse* the MD simulation

e. A set of scripts, stored in the "scripts"directory, that will help us in the automatization of the most time consuming steps.

2.1 Protein simulation

Go to the directory named "lab5".

- PDB preparation: We need to build a PDB file of the protein-ligand complex that `Q` can understand. You will find into this directory the crystal structure of HIVRT with compound **62** (PDB code 2BAN), file `2ban.pdb`. From this file we can extract the coordinates refering to the protein residues. You will also see other PDB files, with the general name `lig`*ligand*`_dk`*docking pose*`.pdb`. Choose your prefered ligand and docking pose and run the script `prepare.sh` giving as argument the code of the ligand (62 or 46) and the docking pose (1, 2, ...)(`tcsh scripts/prepare.sh ligand pose`). The scripts performs some neccesary text editing on the original PDB files, and creates the complex and the ligand files ready to read by `Q`. It is important to understand what has been done, since there are some essential steps in the setup for the calculation in there:

- Shrink a sphere of residues with at least one atom within 20A of the ligand, using pymol (see lab 1)
- A "GAP" line is introduced between molecules, instead the default TER used by PDB convention. Note that when we shrink the protein several “pseudo molecules ”are created.
- Remove the chain ID (5th column of the PDB file). The last two steps correspond to particularities of the program `Q` when reading PDB files
- Ionizable residues (aspartic, glutamic, lysine or arginine) and protonation state of the histidines: in order to ensure a correct solvation of any charged group present in the system, only those ionizable residues located deep in the sphere of simulation will be considered as charged. The OPLS residue library file (`Qoplsaa.lib`) contains both the charged (ASP, GLU, LYS, ARG) and neutral (ASH, GLH, LYN, ARN) version for these residues, so for a given residue changing the default residue name to the charged version name on the pdb will solve the problem. Histidines can be protonated in δ (HID), ε (HIE)or they can bear a full positive charge (HIP). In our case, after visual inspection, all histidines can be modeled as HID while only the following residues within the sphere will be charged:

  * ASP: 191A, 186A, 237A
  * GLU: 138B
  * ARG: none
  * LYS: 101A, 102A, 103A, 172A
- There are three residues for which the cristal structure does not provide the conformation of the relative sidechains. These have been mutated to Ala in order to simplify the protein preparation, since they are far from the binding site: Lys220A, His221A, GLN222A, LYS323B
- Now we are ready to prepare the ligand coordinates and join them (separated by a GAP entry) to the protein coordinates in order to create the complex. The first step is to translate the atom and residue names from the Autodock PDB in a way that is compatible with the ligand library file. The perl script “adk`ligand`2Q.pl ”will do that part. A simple UNIX `cat` command will merge the protein and ligand atoms into a single PDB named lig_p.pdb
- Note that the PDB does not need explicit hydrogens. They will be automatically added by `Qprep5`

After you run this steps (remember, use the script “prepare.sh ”, you will have a directory named lig`ligand`_dk`docking pose`, corresponding to the complex that you have chosen to simulate. Within that directory, you will have the two needed directories named “protein ”and “water ”. Move into the directory `lig`*ligand*_`dk`*docking pose*`/protein` and take a look at the files that you have created in there.

- Run `Qprep5` in order to create the topology. The program can run in interactive mode, giving several commands as arguments, or alternativelly (and most recomended) you can write all `Qprep5` commands in a separate file (i.e. **maketop_p.inp**. Please open the input file (maketop_p.inp) and try to understand the process of `Qprep5`: reading the input library and topology files, reading the pdb file, solvate the system and writing the topology.

run `Qprep5 < maketop_p.inp > maketop_p.log`
Now you can carefully analize the output given by the program.

- Which is the total charge of the simulation sphere?
- Which is the size of the simulation sphere?
- How many water molecules have been added to solvate the system?

Now it is the moment to create the input files for `Qdyn5`. The template input files are stored in the scripts directory, just run the script create_inputs.sh (as usual, stored into the `scripts` directory) giving as argument "protein"(i.e., `tcsh ../../scripts/create_inputs.sh protein`).
The input files that we have just created are consecutive blocks of MD simulations. Any MD simulation with explicit solvent is divided in two phases:

- Equilibration phase: The system must be equilibrated, since we start from a frozen image of the complex (*i.e.* crystallographic coordinates) solvated with a predefined grid of waters.
  In this practical there will be 6 blocks of equilibration, given by the input files `eq1.inp` to `eq6.inp`. The variables of each block are outlined in the following table:

Input file	Starting file	Temp (K)	Bath coupling (fs)	Δ t (fs)	*n* steps	Force constant ($\frac{\text{kcal}}{\text{mol}}\text{Å}^2$)
eq1.inp	complex.top	1	0.1	0.1	5000	25
eq2.re	eq1.inp	50	5	1	5000	10
eq3.re	eq2.inp	150	5	1	1000	5
eq4.re	eq3.inp	310	20	1	10000	2
eq5.re	eq4.inp	310	100	1	20000	1
eq6.re	eq5.inp	310	100	1	50000	0

  * `eq1`: It is similar to energy minimization of the solvent and hydrogens of the solute: in this 0.5 ps run, we use a short time step and a strong coupling to the thermal bath at 1 K temperature, and heavy solute atoms restrained to their starting positions.
  * `eq2 - eq5`: The system is gradually heated to 50, 150 and 310 K during 0.5 ps on each temperature point, with the time step increased and coupling constant relaxed. The restraints on heavy solute atoms are gradually relaxed.
  * `eq6`: Same conditions as the production phase, 50 ps of unrestrained equilibration.
- Production phase: This is the part of interest, which will be later analyzed by extracting the information about the energies and other properties of interest. The MD simulation is divided into 10 consecutive blocks named `dc1.inp` to `dc10.inp`.
  * How long is the total simulation (production phase, in ps)?

  Running the MD simulations would take about 10 hours in a single processor CPU. You can take a look at the script `sub.sh` that was used to submit this job in Finisterrae at CESGA (2h run using 6 processors, with the parallel version of `Qdyn5`). In order to save time the output files have been stored in the directory `results` for each ligand.

2.2 Water simulation

Now move to the directory `lig`*ligand*`_dk`*docking pose/water*. There you will find:

- The coordinates of the ligand `lig_w.pdb` as extracted from Autodock and converted previously with our script `prepare.sh` (but with all the protein atoms removed).
- Run `Qprep5`as you have done with the "protein "simulation. (`Qprep5 < maketop_w.inp > maketop_w.log`. The topology generated by `Qprep5` will consist on the ligand positioned in the center of a sphere of radius 18Å filled with water molecules, the input files and the `fep` file can be obtained exactly as in the case of the protein simulation (`tcsh ../../scripts/create_inputs.sh water`). All files are named according to the same convention as for the protein simulation.

As previously the MD simulation is divided into two phases.

- **Equilibration phase**: The equilibration phase (`eq1.inp` to `eq5.inp`) is similar to the protein simulation, but one important change has been made. Since there is no protein present in the simulation, only a restraint that keeps the center of mass of the ligand in the sphere center has been added:

```
[sequence_restraints]
1 53 10 0 1
```

- **Production phase**: The production phase is basically the same as with the complex. The restraint that keeps the ligand in the center of the sphere is maintained to ensure a correct solvation of the ligand.

Again, running the MD simulations would take about 10 hours. In order to save time the output files have been prepared in the directory `results`.

3 Results: Evaluating the simulations

Now we will evaluate energies from the MD simulations, the conformations of the system and specific ligand-protein interactions.
The results of the MD simulations are stored under `lab5/results` directory, in the form of a compressed tar gzipped file. Do the following steps to retrieve the results corresponding to a given ligand-docking pose:

- Move to the directory `lig`*ligand*`_dk`*docking pose* (under the parent `lab5` directory).
- Copy the corresponding tar file: `cp ../results/lig`*ligand*`_dk`*docking pose*`.tgz ./`
- Uncompress the file: tar -zxvf `lig`*ligand*`_dk`*docking pose*`.tgz`

Great! You have stored the output files in the corresponding `protein` and `water` directories. Lets proceed with the analysis

3.1 Energies

- **Convergence:** First we will look at the convergence of the simulations, in particular what is referred to the ligand-surrounding resultat for the ligands in water and protein simulations. By using the `perl` script `resultat.pl`, the *electrostatic* and *van der Waals* energies are extracted from the `name.log` files.
  - Move to the `protein` directory corresponding to the ligand of interest (`lig`*ligand*`_dk`*docking pose/* directory)
  - Open a `dc?.log` file and try to find the ligand-surrounding energies for the water and protein simulations.
  - Now run the script `perl resultat.pl 1 10`, which extracts, summarizes and plots all ligand-surrounding energies for the entire production phase (`dc1.log` to `dc10.log`).
  - ADVANCED USERS. The perl script is very powerful, type `perl resultat.pl -h` to check how it works and the different options. Note that if you do not consider the trajectory converged, you can discard a given part of the MD simulation by selecting the appropiate file indexes (i.e. perl resultat.pl 6 10 will only consider the second part of the simulation). Also note that the script calls the gnuplot program. You can replot at any time you like the energies with gnuplot, which are stored in the file `plot.txt`, and you can check the averaged values on the file `resultat.txt`.
    - Are there any large changes in the energies throughout the simulations?
    - The script has also calculated the difference between the average energies for the first and second halves of the simulation. This can be considered as a measure of the convergence error. Have the simulations converged?
  - Repeat this procedure for the corresponding water simulation and summarize the ligand-surrounding energies in Table 1.
  - Proceed with a different ligand or copy the results from a colleage in the lab.
- **Binding free energies:** The calculated ligands-surrounding energies can be used to estimate the free energy of binding.

Ligand	pose	$\langle V_{l-s}^{vdw} \rangle_{\mathrm{p}}$	$\langle V_{l-s}^{el} \rangle_{\mathrm{p}}$	$\langle V_{l-s}^{vdw} \rangle_{\mathrm{w}}$	$\langle V_{l-s}^{el} \rangle_{\mathrm{w}}$	err_p^{vdw}	err_p^{el}	err_w^{vdw}	err_w^{el}
62									
46									

Table 1: Ligand-surrounding energies for ligand 62 and ligand 46

- Calculate the LIE binding free energies using the parameterization; α=0.18, β=0.43 and γ = -10.2 and summarize these into Table 2.
- Are the calculated binding free energies in agreement with experimenal data? Are the ligands correctly ranked?
- Is the binding due to electrostatic or hydrophobic interactions?

Ligand	$\Delta\langle V_{l-s}^{vdw} \rangle$	$\Delta\langle V_{l-s}^{el} \rangle$	$\Delta G_{bind,calc}$	err_{calc}	$\Delta G_{bind,exp}$
62					-12.8
46					-5.7

Table 2: Differences in ligand-surrounding energies and calculated free energies using LIE

3.2 Structures

Viewing the structures after the different parts of the simulation is a very important part of the evaluation of the simulations.

- Move to the `protein` directory corresponding to the ligand of interest.
- To look at how the complex evolves as a function of time, we will extract the snapshots from the `Q` restart files `dc?.re` using `Qprep5` and create a `PyMOL` script. Use the script `structures.sh` under the `scripts` directory. You will see a frozen image of the starting structure (top.pdb) and as long as you press the "play "button (or the right and left arrows), the snapshots corresponding to ps 50, 100, 150 ...
- ADVANCED USERS: Other ways exist to look at the structural convergence of the simulations.
  - It is also possible to calculate average structures from a given frame of the MD simulation with `Qcalc5`. The commands for `Qcalc5`are stored in the files `ave?.inp`. Type `Qcalc5 < ave1.inp` to get the average of the first half of the trajectory (or use `ave2.inp` for the average of the second half). You can load the resulting files (ave1.pdb and ave2.pdb) together with the initial structure (top_p.pdb), and check the structural convergence of the simulation.
  - Additionally, you can evaluate the RMSD of the ligand (or selected residues) with `Qcalc5`. Use `Qcalc5`with the input file `rmsd.inp` (in this case, the whole trajectory is considered and only the RMSD of the ligand is calculated). The results are stored in the file `Entropy.out`. Plot that file with gnuplot: is the ligand stable in the binding site along the simulation?
  - You can load the complete trajectory files (`dc*.dcd` in pymol, using the `dcd loader` plugin and loading previously the PDB file mask_dcd.pdb. Consult the teacher for doing that, probably it is not possible in the installed version of `PyMOL`

3.3 Key interactions

To elucidate which residues contribute most to ligand binding, the energetic interactions of the ligand with its surrounding groups can be calculated. This can be done using the program `Qcalc5`.

- Be sure that you continue in the `protein` directory corresponding to the ligand of interest.
- type `Qcalc5 < res_Q.inp > res_Q.log`. Rescue the last 202 lines of that file, (i.e. `tail -202 res_Q.log > res_Q.txt`), since each line corresponds to each residue in the protein, and the columns mean resID (1st column), $\langle V^{vdw}_{l-resID} \rangle_{\mathrm{p}}$ (2nd column) and $\langle V^{el}_{l-resID} \rangle_{\mathrm{p}}$ (3rd column), you can plot this file in `gnuplot`. You can easily identify the residues that play an active role in ligand binding.
- You can also visually inspect the location of this residues in the complex structure. The script `res_imp.sh` under the `scripts` directory will create and run a pymol session similar to the one created in section 3.2, but highlighting the residues that do electrostatic or non-polar interactions with the ligand within a minimum threshold of $\pm 3 kcal/mol$. The residues with non-polar interactions are depicted in dots, while the residues contributing to the electrostatic component of the free energy of binding are colored yellow.
- Which residues contribute most to the binding of your ligand?