Implementation of a Knowledge-Based Scoring Function

Thanh Lai
4th year Chemistry Undergraduate

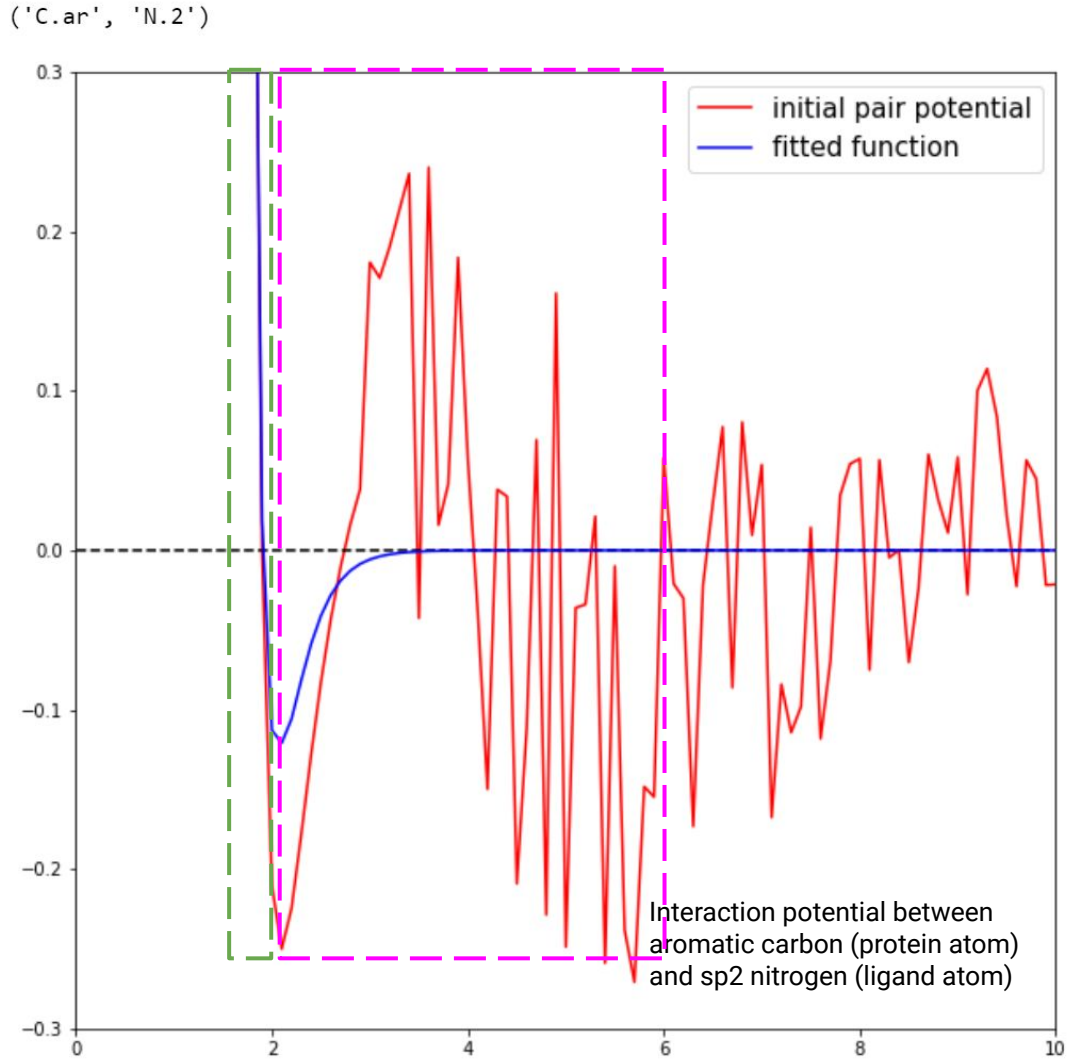
Drug Discovery and Scoring Functions

- The project goal is to implement a python codebase of a scoring function called **ITScore**
- Scoring functions are fast algorithms that qualitatively evaluate the **stability** of a protein-ligand complex
- A knowledge-based scoring function uses crystallographic data to derive pairwise **protein-ligand atomic potentials**
    - Uses a **set of statistical potentials** to evaluate a complex's stability
- In a **drug screening process**, thousands of candidate molecules are docked onto a target protein and their binding affinities are evaluated in a high throughput manner

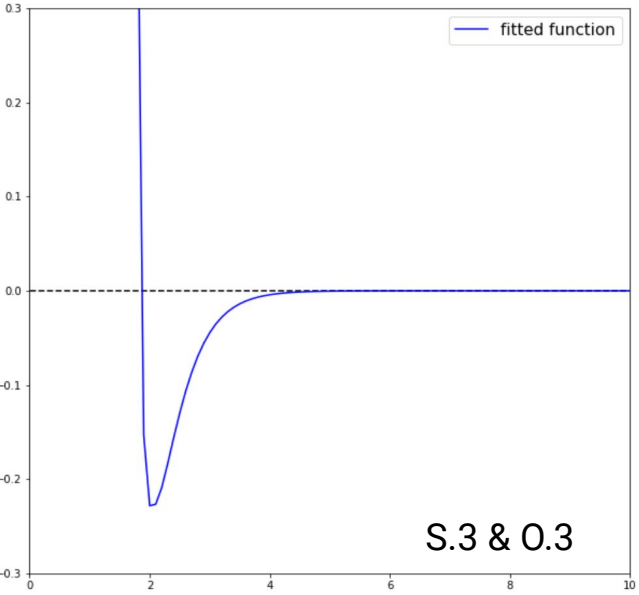
Most potentials are fitted quite nicely

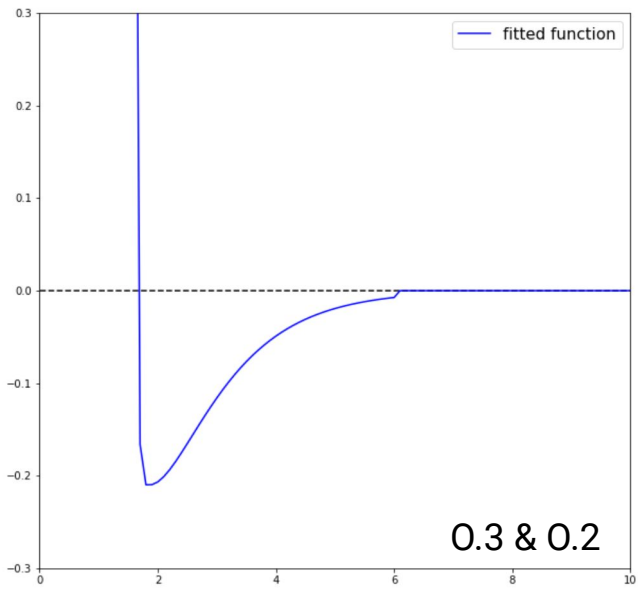
$$F(r) = \begin{cases} \kappa(r_{\min} - r)^s - \epsilon_0 & r < r_{\min} \\ \epsilon_0(e^{-\alpha(r-r_{\min})} - 1)^2 - \epsilon_0 & r_{\min} \le r \le r_c \\ 0 & r > r_c \end{cases}$$

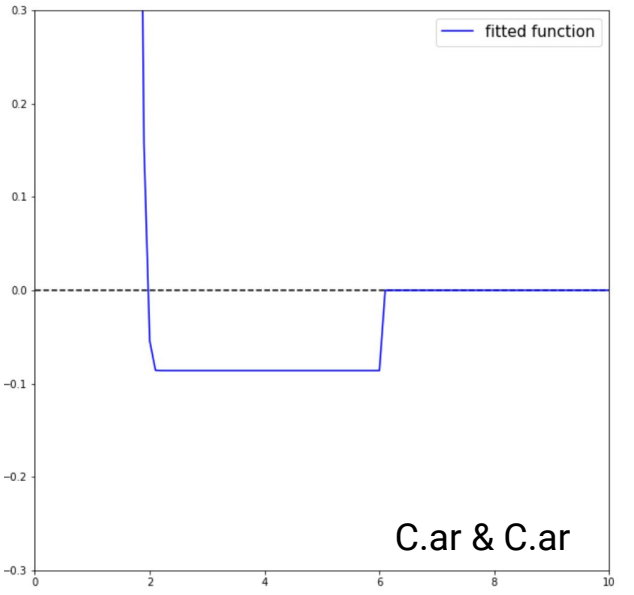
Eq 1. Smoothing function used.
First function is VDW repulsion
Second function is morse potential
Third is zero interaction



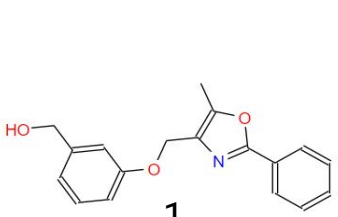
Cool (and not so cool) looking potential functions



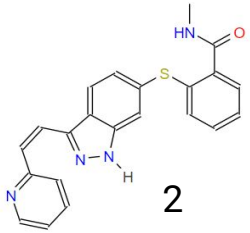




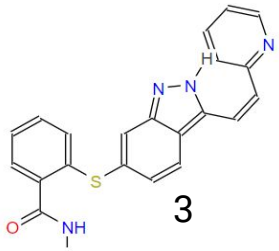
How well does it score protein-ligand complexes?



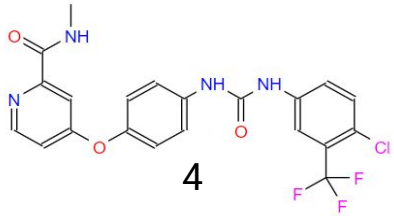
1



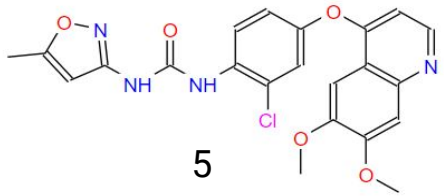
2



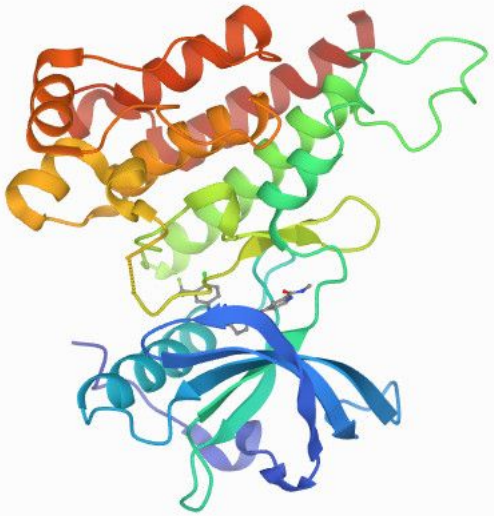
3



4



5



Ligand	Score (Unitless)	Experimental Dissociation Constant (M)
1	-7.09	25e-6
2	-7.65	1.1e-9
3	-7.71	0.02e-9
4	-8.54	0.52e-9
5	-9.70	0.01e-9
Kendall-Tau	0.80	

Things I learned

- How to use Biopython
- How to compile, install, and use OpenBabel
- Some statistical mechanics (radial distribution function, potential of mean force, etc.)
- Methods of storing functions in python
- How to fit a curve to a piecewise function

Reflections & Improvements

- Did not utilize my time efficiently so I could not implement the iterative training part
  - Either way, it seems to rank ligands quite well with the “crude” potentials
- PDB files do not contain atom types I needed, so if I started this again I would immediately use OpenBabel instead of wasting time with PDB files
- I originally implemented one of the equations wrong which led to weird looking potentials. Next time I will be more careful with reading the paper & equations
- Implement this with C++
  - The scoring will be faster and I would have a reason to relearn C++, although this will make the project much harder

Acknowledgements

- Arjun (green-lighting the project)
- Eva (peer reviewer)