BOSS Output Filtration

Methods

Our goal is to filter out all same or “similar” molecules, so the question becomes how do we define “similar” molecules?

Molecule is constructed by atoms, which again can be represented by their coordinates. To unambiguously define a molecule’s configuration in 3D space, the Degree Of Freedom for linear molecule is 3N-5 and 3N-6 for non-linear molecules, where N means number of atoms.

Focus back on our problem, for conformations from BOSS outputs, they are well constrained during the simulation, in other words, the molecules as the input will be still as the molecules accordingly in a whole, it is unlikely they can be broken into pieces unless we specifically set to. Thus, we trust BOSS constrains.

With the knowledge that the molecule breaking the strict constraints will have large penalty functions, the good constrains will make most molecules appear similar or without big changes. So, we do not need to waste time on their self-filtrations.

As a summary, we only need to perform filtration on molecule-molecule combinations.

Talk back to the definition of “similarity”, we will use two types of constrains, e.g., bonds & angels.

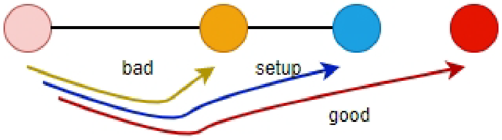


Figure 1: Definition of Bonds & Tolerance

Assume there are many molecules, we label one of those molecules as the reference, then the difference between this reference’s first two atoms as the Bonds-Length-Reference. Calculating the difference for other molecules accordingly, we will get the sets of Bonds-Length-Others.

Now, we can calculate the difference between any sets in Bonds-Length-Others and Bonds-Length-Reference, a new set will be got, which is labelled as Bonds-Length-Variance.

Performing the same calculation for other molecules when they are as the reference, then all the calculation results will be labelled as the Bonds-Length-Variance.

For the Bonds-Length-Variance, we can setup a Bonds-Length-Tolerance, showing in Figure 1, so any values are smaller than this number, their combinations will be marked.

For example, we have four molecules, [1, 2, 3, 4], then we can calculate the Bonds-Length-Variance for molecule combinations, 1-2, 1-3, 1-4, 2-3, 2-4.

Finally, we can do filtration on those marked combinations. A technical question arises, we want to keep as many molecules as we can when we are doing filtrations.

For example, we marked molecule combinations 1-2, 1-3 as the similar repeats, due to molecule combination 2-3 is not, so we only need simply remove molecule 1, then everything will be perfect fine.

This is the core part from code view to identify the molecule in similar repeats, we define them as Molecule-Bonds-Repeats.

I will not go into detail about the methods used for codes, everything is explained inside.

Similarly, we can do calculation for angles. They are defined as Angles-Degree-Reference, Angles-Degree-Others, Angles-Degree-Variances, Angles-Degree-Tolerance, Molecule-Angles-Repeats, respectively.

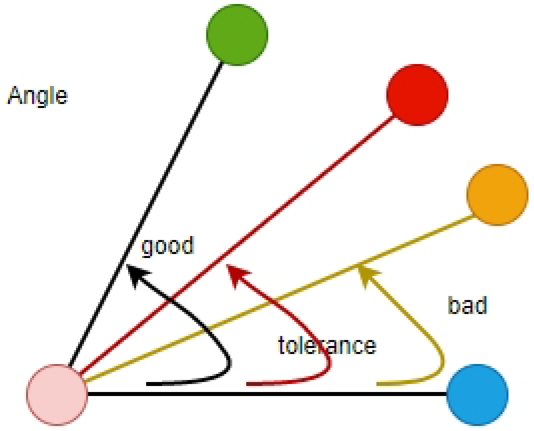


Figure 2: Definition of Angles & Tolerance

Because whether molecule changes on bonds or angles, they all can be thought as molecule in changes. Therefore, we only need to filter out molecules with no changes, neither they are on bonds or angles.

Thus, we can do a simple search on repeats in Molecule-Bonds-Repeats and Molecule-Angles-Repeats, we label those molecules as Molecule-Repeats-Hard. Removing them firstly, then using the Code Core Part filter out molecules either in Molecule-Bonds-Repeats or Molecule-Angles-Repeats, we label them as Molecule-Repeats-Soft.

Both Molecule-Repeats-Hard and Molecule-Repeats-Soft are molecules in repeats, they all should be removed.

Because we think the BOSS input as a single big molecule, so for the molecule not linked with any atoms will be thought as fragments, the same meaning as “molecules”. To differentiate, we say the BOSS input in a whole as the one big molecule, any independent pieces or residues will be thought as fragments. Thus, we do calculation on the cross-comparison of difference among those fragments, which has been explained in above.

Now the question becomes, how can we detect those fragments? In script, it provides two type of identifications.

1) By user inputs

For example, if we have a BOSS input like,

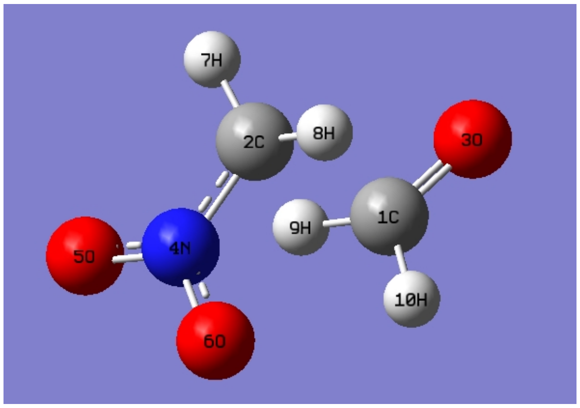


Figure 3: BOSS input

We will know, it has two fragments, distinct from atom index, they are fragment A: [2,4,5,6,7,8] & fragment B: [1,3,9,10].

Because we trust BOSS constrains, we only need to calculate difference between fragment A and B. Thus, we only need to calculate bonds difference between atoms [2-1, 2-3, 2-9, 2-10], angles differences [2-4-1, 2-4-3, 2-4-9, 2-4-10].

Or any combinations that user can customize.

2) Auto identification

Codes use the following two references:

a) Wang, J.; Wang, W.; Kollman, P. A.; Case, D. A., Automatic atom type and bond type perception in molecular mechanical calculations. *Journal of Molecular Graphics and Modelling* 2006, 25 (2), 247-260.
b) Zhang, Q.; Zhang, W.; Li, Y.; Wang, J.; Zhang, L.; Hou, T., A rule-based algorithm for automatic bond type perception. *Journal of Chem informatics* 2012, 4 (1), 26.

Assumptions

In general, they are two assumptions, both of them can be used as the circumstantial evidences to check the validation and efficiency of BOSS constrain algorithms, as well as the direct evidence for our suspect.

First assumption: molecule distribution

We suspect we have over-trained many molecules in similar. As a consequence, 1) those molecules are in a higher ratio than any other molecules in our training sets, which will heavily influence training equalities, thus as a result causing our training program, e.g., AENET, will have the tendency to make prediction mostly based on those favorable conformations; 2) they cost much unnecessary time and efforts.

We assume if we plot those molecules out, we will see a higher ratio of those molecules in similar. Besides, due to the good BOSS constrain algorithm, those higher ratio molecules should be in the middle, the plot should be like Gaussian Distribution or Normal Distribution, in mathematics.

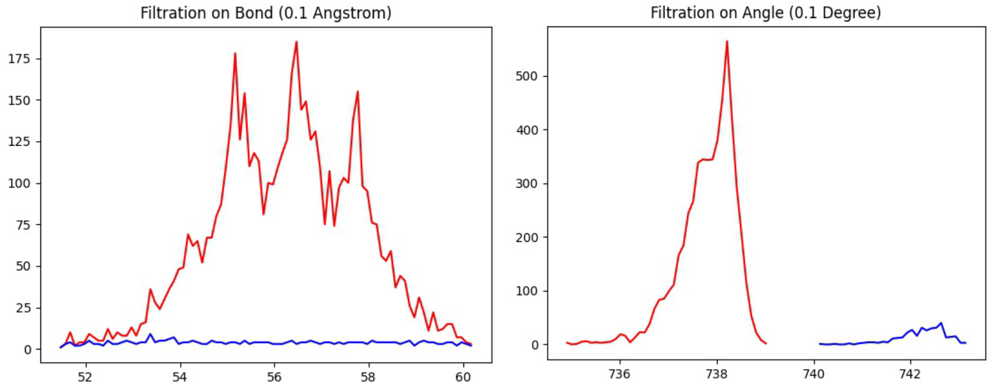
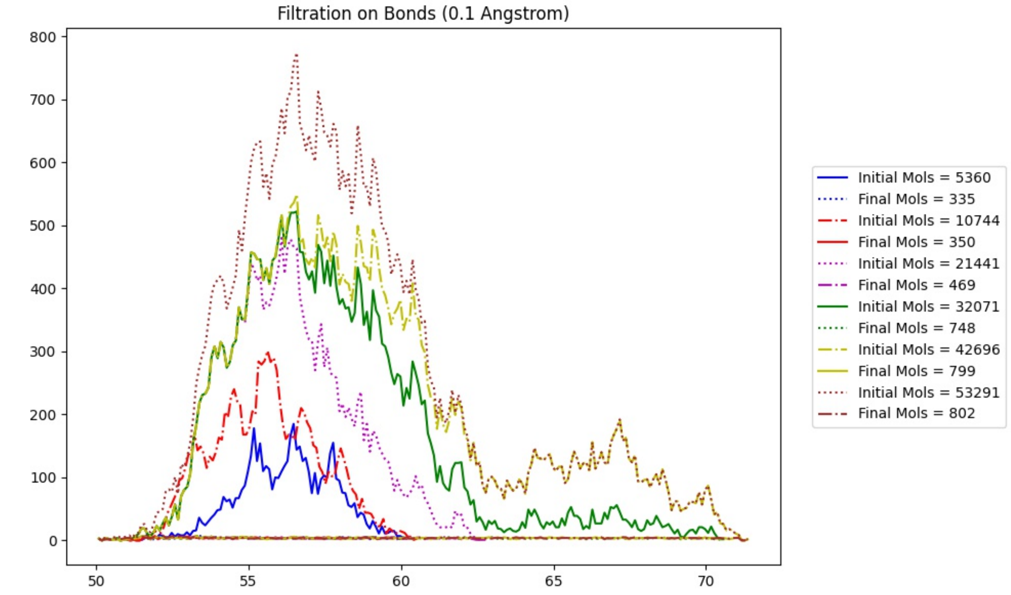


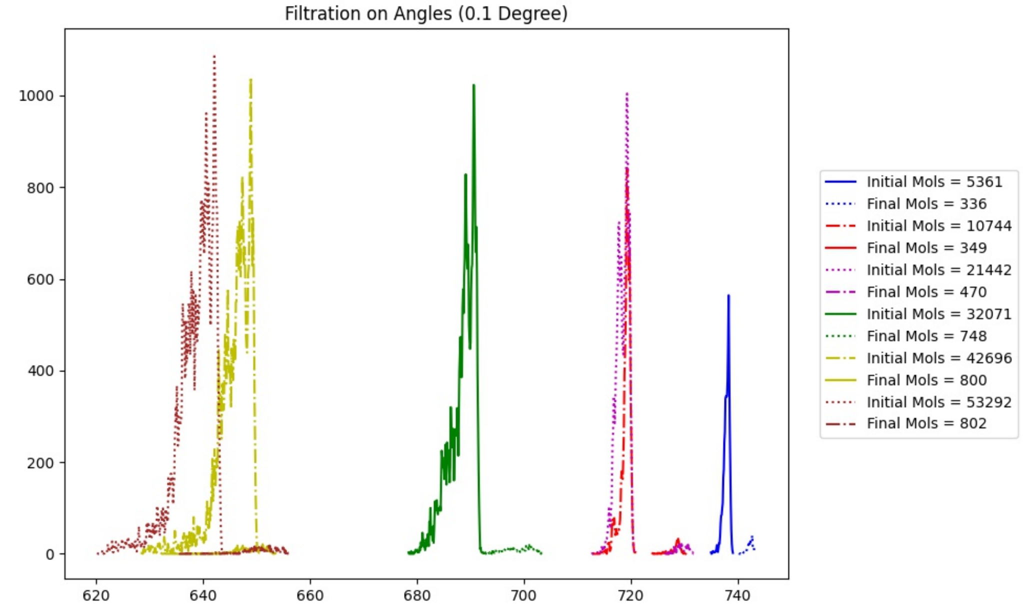
Figure 4: Plot For Bonds & Angles Filtrations. (Before: red; After: blue. On 5361 Molecules)

From Figure 4, it exactly shows what we think.

Second assumption: distribution as number of molecules increase

We assume BOSS has good constrain algorithm, so the ratio and probability of BOSS outputs for favorable molecules should be larger than the molecules are unfavorable. Thus, as the number of inputs increase, the plot should be more and more like ideal Gaussian Distribution.





Because the filtration is firstly performed on bonds, following with angles. Then we only need to focus on plot on bonds. This again supports our guess.

Besides, it also supports that if we have an infinite number of molecules as inputs, finally the plot will be shown as a single peak line in the middle.