

# report\_buildH

August 14, 2019

## 1 Report on buildH hydrogen rebuilding and order parameter calculation

Author: Patrick Fuchs (August 2019)

### 1.1 Summary

In this document are reported different elements judging the accuracy of **buildH** for calculating and building hydrogens on a united-atom (UA) trajectory.

As a reminder, so far we have three programs:

- The [program from Josef Melcr](#): it calculates the order parameter (OP) from a trajectory *with hydrogens*. It has now replaced some awk scripts used in NMRlipids I. In the following I'll call this program **JMelcrProg**.
- The [program from Angel Pineiro](#): it rebuilds hydrogens and calculate the OP from a united atom trajectory. There's also an option `IGNH` which takes as input an all-atom trajectory and ignores the hydrogens there; in this case hydrogens are rebuilt and the OP are calculated on these latter. In the following I'll call this program **APineiroProg**.
- The [program \*\*buildH\*\* from Patrick Fuchs and co-workers](#): **buildH** builds hydrogens and calculate the OP from a UA trajectory. It has also an option to output a trajectory with the rebuilt hydrogens in xtc format.

Here we use a trajectory generated with the CHARMM36 force field (links to [github](#) and [zenodo](#)) and test the 3 programs. The system has 256 POPC and the time window analyzed is 50-300 ns (one frame every 100 ps), thus on 2500 frames. For easier comparison, **buildH** produces two outputs in a format similar to both **JMelcrProg** and **APineiroProg**. In the following sections, I systematically put links to the files with numerical values in case they are useful.

### 1.2 Test on one frame of 256 POPC

In this section, we calculate the OP on a single frame (the first one of the trajectory).

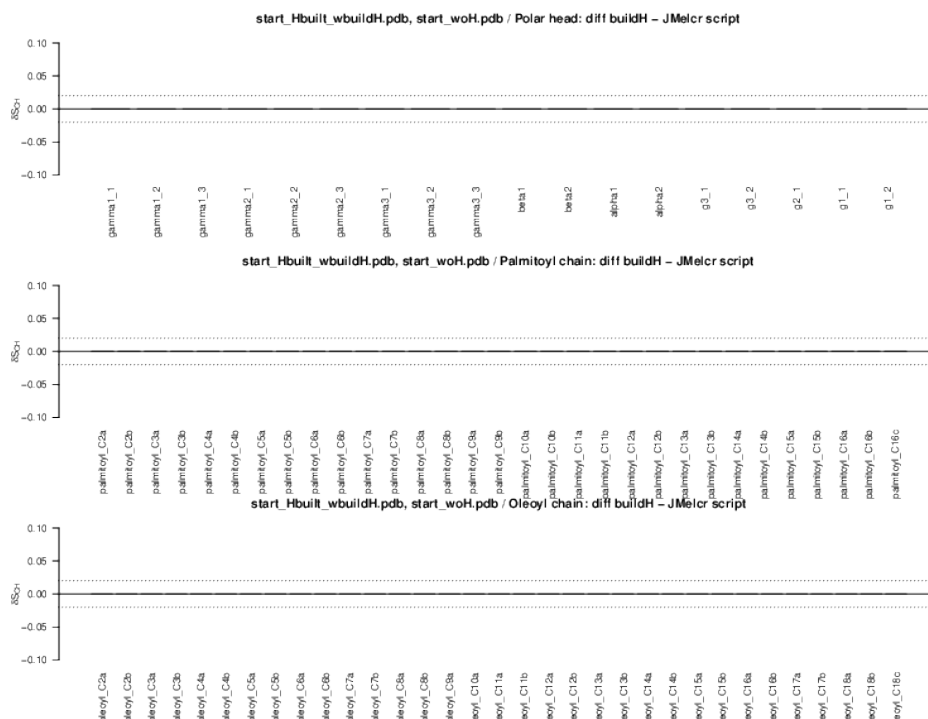
#### 1.2.1 OP calculation

To check that **buildH** calculates correctly the OP, I did the following:

1. I took a single frame from the CHARMM trajectory (1st frame): `start.gro` ([file](#)). Thus, this frame has all the Hs.

2. I removed the hydrogens and got start\_woH.pdb ([file](#)).
3. I re-built Hs on start\_woH.pdb (and calculate OP) using **buildH** and obtained start\_Hbuilt\_wbuildH.pdb ([output](#)).
4. I launched **JMolcrProg** with start\_Hbuilt\_wbuildH.pdb ([output](#)).

I compare in the next barplot the OPs obtained in step 3 and 4. The  $y$ -axis is the difference of OP:  $\delta S_{CH} = S_{CH}(\text{buildH}) - S_{CH}(\text{JMolcrProg})$ . The two dashed line indicate a difference of  $\pm 0.02$  in units of OP. All the remaining plots in this document follow the same rules. **Beware**, if the difference is higher than 0.1 (or lower than -0.1) it may go out the plot area.



Difference between JMolcrProg and buildH on OP calculation

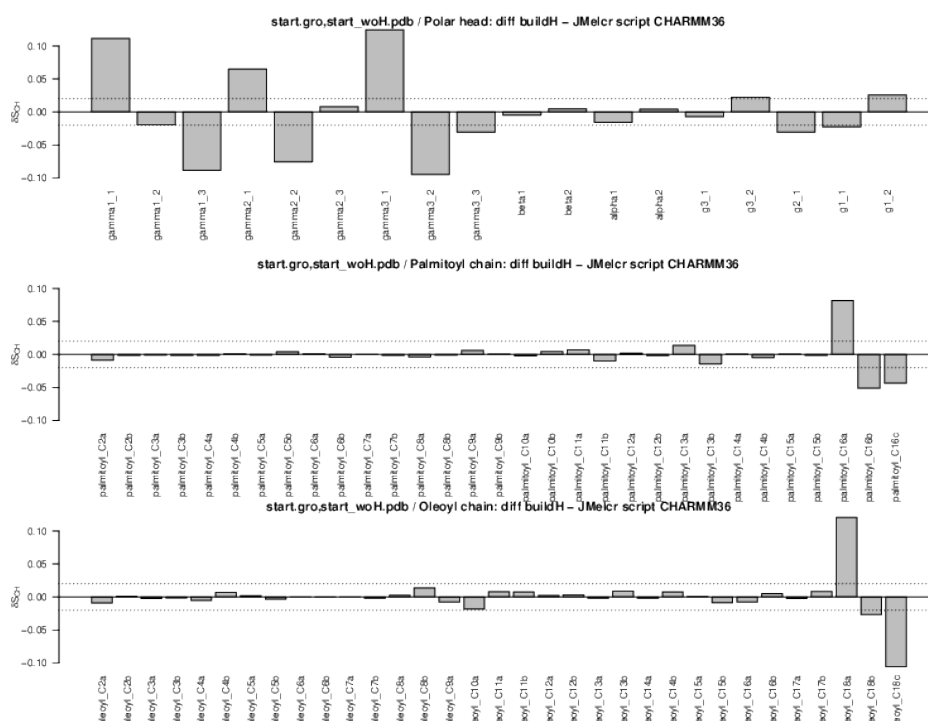
With this protocol, I can check whether the two programs agree on the OP calculation, using the exact same coordinates (up to the 3rd digit in the PDB). The mean absolute value difference is  $1.33\text{e-}05$  meaning that all OPs are indistinguishable. Thus we can say that **buildH** and **JMolcrProg** fully agree on the calculation of OP from a given C-H position.

## 1.2.2 Accuracy of buildH vs AA traj from CHARMM36

Here I compare the OP calculated with:

1. **JMolcrProg** on start.gro, thus we have the "real" OP calculated from a CHARMM36 all-atom (with Hs) structure ([output](#));
2. **buildH** on start\_woH.pdb, thus with Hs rebuilt from the heavy atoms' position ([output](#)).

In the end, what we compare in the next figure is the real OP from an AA trajectory, with that from rebuilt Hs using **buildH**.

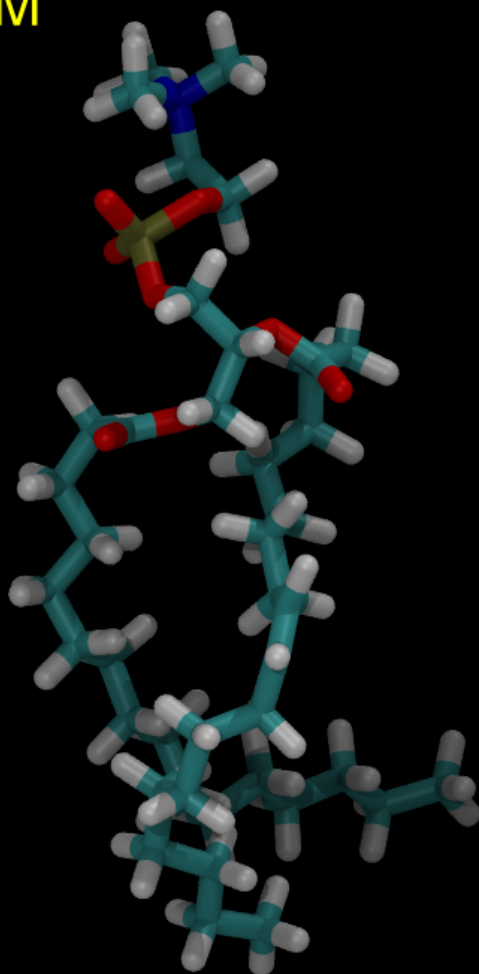


Difference between JMolcrProg (CHARMM36) and buildH on OP calculation

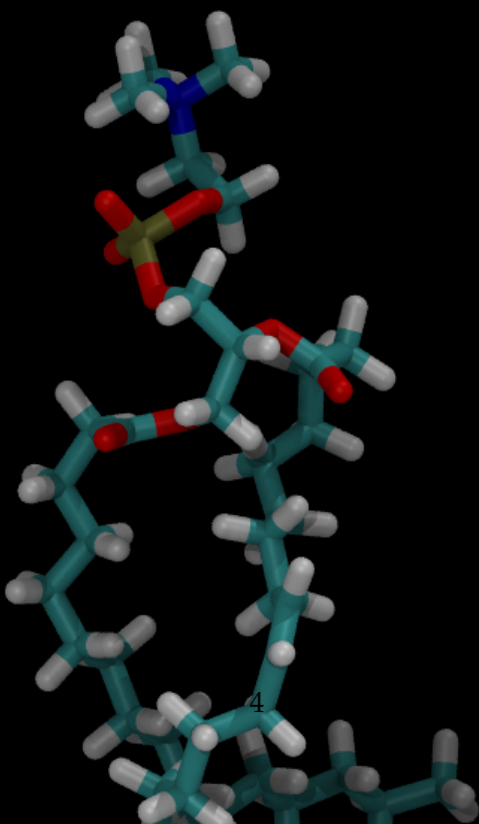
The mean absolute difference is 0.018, but there is a great variability between the different Hs:

- First, we have a high difference on all methyl (CH<sub>3</sub>) groups, sometimes higher than 0.1 unit. This is expected since during the construction of the 3 Hs with **buildH**, we have to choose an azimuth for the first H (let's call it *newH1*) that we rebuild (the other 2 are deduced from this one). By default, this is chosen opposite to the two helpers, so that the dihedral *newH1-carbon-helper1-helper2* is *trans*. For the AA trajectory, the 3 Hs can freely rotate. So the probability to find the same azimuth for *newH1* is very low. This tells us that we should compare the average over the 3 Hs since we cannot guess in advance the value of the dihedral *newH1-carbon-helper1-helper2*. One could expect that this difference should decrease if we average over a trajectory.
- The agreement for the two Hs (C9a and C10a) of the double bond in the oleoyl chain is pretty good (-0.008 and -0.018). As mentioned in the [paper from Thomas Piggot](#), using the bisection of the C-C-C angle - as is done in **buildH** - leads to a good approximation of the position of these Hs.
- There is a difference that slightly exceeds 0.02 units on 4 out of 5 Hs of the glycerol. The animated gif below shows that assuming ideal geometry results in differences on all Hs. Averaging over a trajectory will somehow reduce those differences on most of the Hs (see below). However, the glycerol Hs probably deviate on average from ideal geometry because of some subtle effects from the neighboring oxygens. Again, we expect to lower this difference by averaging over a trajectory.
- Last, for all the other Hs (choline, tails, etc.) **buildH** agrees very well with AA results.

CHARMM



buildH



Some minor details :

- In general, the calculation of stdev and STEM agree very well between both progs. Here's an example on some Hs from start.gro (the 3 cols of floats correspond to the OP, stdev and STEM respectively):

Example taken from **buildH**:

beta1	POPC	C12	H12A	-0.03356	0.42328	0.02646
beta2	POPC	C12	H12B	-0.06606	0.41748	0.02609
alpha1	POPC	C11	H11A	0.01963	0.45025	0.02814
alpha2	POPC	C11	H11B	0.07456	0.46563	0.02910

Example taken from **JMelcrProg**:

beta1	POPC	C12	H12A	-0.02880	0.42511	0.02657
beta2	POPC	C12	H12B	-0.07085	0.40932	0.02558
alpha1	POPC	C11	H11A	0.03525	0.45881	0.02868
alpha2	POPC	C11	H11B	0.07028	0.46975	0.02936

### 1.2.3 Difference between buildH and APineiroProg

Here I compare the OP calculated with:

1. **APineiroProg** on start\_woH.gro, thus with Hs rebuilt from the heavy atoms position ([output](#));
2. **buildH** on start\_woH.pdb, thus with Hs rebuilt from the heavy atoms position ([output](#)).

Recall that this snapshot has 256 POPCs. The following figure will tell us how similar / different both programs rebuild Hs. Notice that the x-axis labels are atom names (default output from **APineiroProg**).



HS	0.07239	0.46382	0.02899
AVG	0.04549	0.22830	0.01427

I didn't check any further the origin of this.

### 1.3 Test on a full trajectory

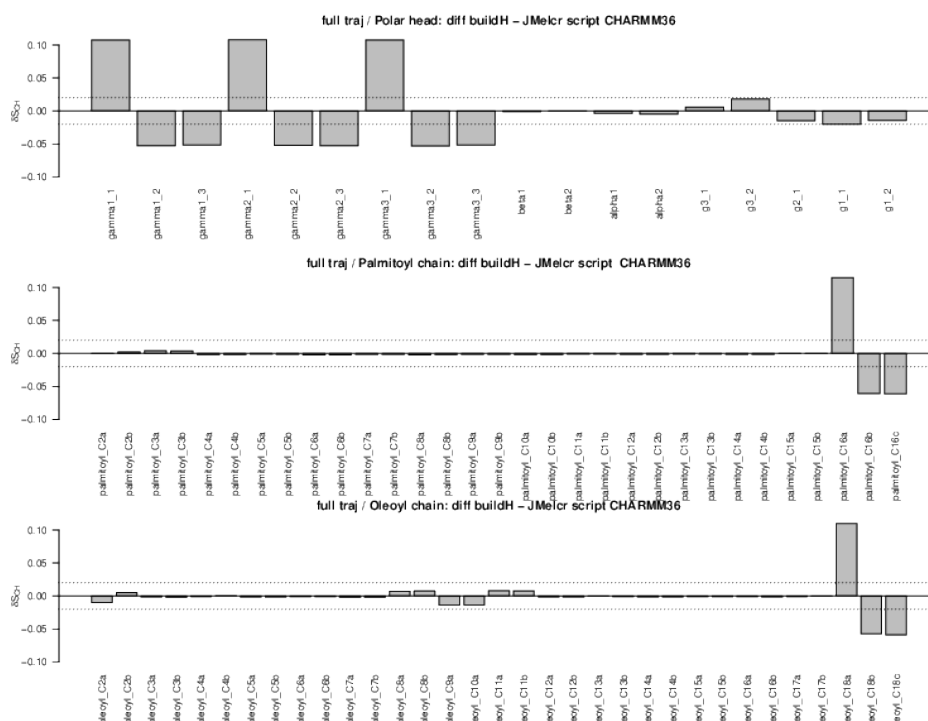
Here I tested the OP calculation using the 3 programs on the full trajectory described above (with 2500 frames).

#### 1.3.1 Accuracy of buildH vs AA traj from CHARMM36

Here I compare the OP calculated with:

1. **JMolcrProg** on traj\_P0PC\_50\_300ns.xtc (extracted from [md\\_dt100\\_0K\\_centered.xtc](#) on the time window 50-300 ns, one frame every 100 ps, 2500 frames), thus we have the “real” OP calculated from a CHARMM36 all-atom trajectory ([output](#));
2. **buildH** on traj\_P0PC\_woH\_50\_300ns.xtc (same trajectory but without hydrogens), thus with Hs rebuilt from the heavy atoms position ([output](#)).

Shown in the next figure is the difference between **JMolcrProg** and **buildH**:



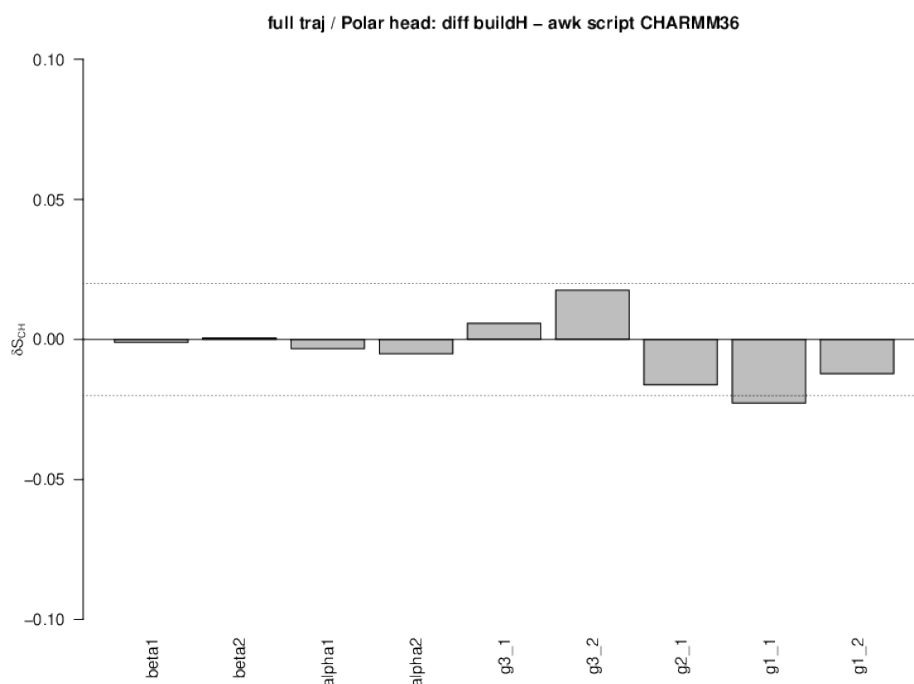
Difference between **JMolcrProg** and **buildH**

Overall, we can say the following:

- The mean absolute difference is now 0.016. As expected, the differences are “diluted” compared to above (on a single PDB);

- The CH3s show a prominent difference ( $> 0.1$ ) systematically for the first H (HR), but less for the other two Hs;
- On the glycerol Hs, the difference is systematically below 0.02, which is good (and reassuring!). This shows that indeed averaging over a trajectory slightly improves the agreement. But the difference is still close to +0.02 or -0.02 for g3\_2 and g1\_1. As stated above on a single PDB, I guess we face here the difficulty of building these Hs using ideal geometry vs a force field which takes many other effects into account that we cannot reproduce with a simple geometric rule. In [Thomas Piggot's paper](#), there is an interesting try towards this direction for the oleoyl double bond. They measure the averaged C-C-H angles in the CHARMM36 trajectory and use that value for reconstructing the two Hs (see Figure 5b, blue line manual angle). This procedure improves the agreement compared to other tools. This shows that learning procedures might be a solution to improve the agreement between UA and AA trajectories. However, we have to keep in mind that we have to do it for each force field (FF), and except for validation, there's no point to use reconstructed Hs if we have the AA trajectory!
- On all CH2, the agreement is excellent.

Since there were values on [github](#) produced with the awk script, I could also do the comparison awk script / **buildH**:



Difference between the awk script and **buildH**

The conclusions are essentially the same as for **JMelcrProg**, except maybe for g1\_1 which here displays a difference slightly lower than -0.02. I guess it comes from the difference of implementation or rounding between the awk script and **JMelcrProg**.

On the whole, the reconstruction with **buildH** yields results very similar to the real

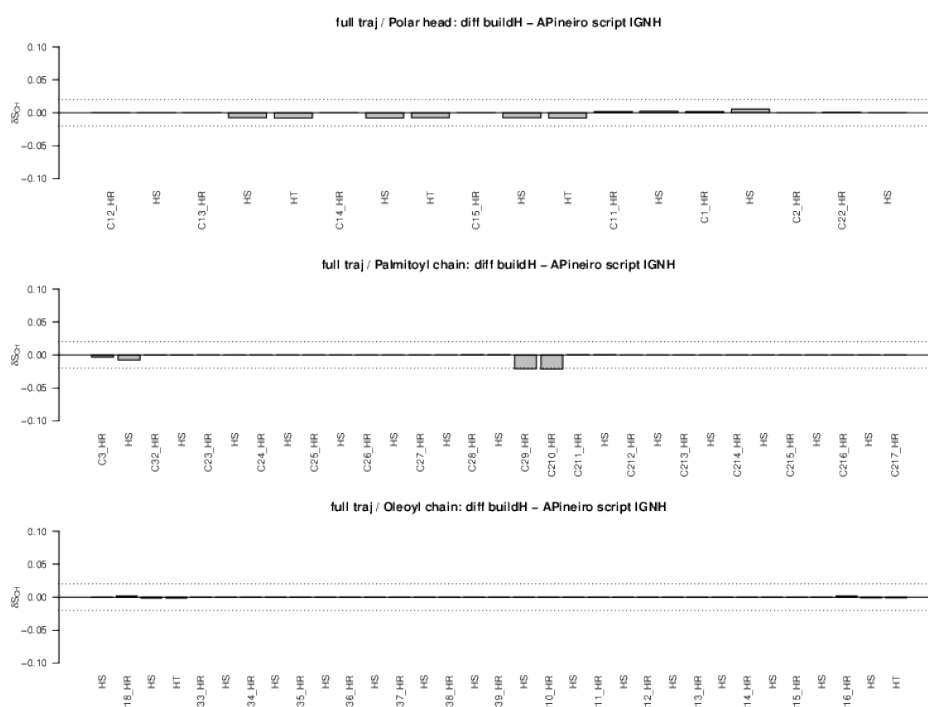


CHARMM36 trajectory with an absolute OP difference no higher than 0.02. This validates the H rebuilding with **buildH**.

### 1.3.2 Difference between **buildH** and **APineiroProg**

Here I compare the OP calculated with:

1. **APineiroProg** on `traj_P0PC_woH_50_300ns.xtc`, thus with Hs rebuilt from the heavy atoms position ([output](#));
2. **buildH** on `traj_P0PC_woH_50_300ns.xtc`, thus with Hs rebuilt from the heavy atoms position ([output](#)).



### Difference between **ApineiroProg** and **buildH**

Again, both programs agree very well except on the two Hs of the oleoyl double bond (difference around -0.02).

### 1.3.3 OP plots

To get another view of **buildH** validity, here are the classical OP plots which compare **buildH** with real CHARMM36 (as assessed by **JMolcrProg**):

