

GROMACS - Feature #1964

"pull=no" should not produce warnings about "unknown" pull keywords

05/18/2016 01:52 PM - Semen Yesylevskyy

| | |
|--|--------------------------------|
| Status: | Accepted |
| Priority: | Normal |
| Assignee: | |
| Category: | preprocessing (pdb2gmx,grompp) |
| Target version: | |
| Difficulty: | uncategorized |
| Description | |
| If "pull=no" is specified grompp produces warnings about each line with pull-related parameters: | |
| WARNING 1 [file md.mdp, line 83]: Unknown left-hand 'pull-nstxout' in parameter file ... | |
| This is inconsistent with other keywords. For example "Pcoupl=no" does not warn about undefined tau_p. The purpose of "pull=no" is to switch off pulling quickly without deleting whole pull block from mdp file, thus grompp should silently disregard all pull keywords instead of producing warnings about them. | |

History

#1 - 05/18/2016 09:33 PM - Chris Neale

maybe... I agree with the consistency logic, but what if somebody inputs all of the pull-code options and forgets to set pull=yes -- then they silently don't get what they expect. I'd be more in favour of going in the opposite direction and throwing an error when tau_p was explicitly defined but Pcoupl=no

#2 - 05/19/2016 02:29 PM - Semen Yesylevskyy

Throwing an error is too strict and not convenient for the user at all. For example I regularly use Pcoupl=no to equilibrate the system with too large forces and then switching it on when the system is relaxed a bit. Removing all pressure keywords and putting them back every time would be very cumbersome.

What about showing a note (not a warning or an error) like "NOTE: It seems that you specified pull-related keywords but the pulling is off. Please double check if this is what you want to do"?

The same message could be shown for Pcoupl=no.

This is consistent, convenient and informs the user about suspicious combination of options at the same time.

Another option is to throw a single warning like "WARNING: pull-related keywords specified while pulling is off!". 100+ warnings about many pull groups is way too much. However, I prefer a note since this combination is valid and thus should not abort grompp.

#3 - 05/19/2016 05:13 PM - Chris Neale

You might keep base parameters in one .mdp file and pull parameters in another .mdp file then build the proper .mdp file before grompp.

```
gmx grompp -f base.mdp
gmx mdrun ...
cat base.mdp pull.mdp > this.mdp
gmx grompp -f this.mdp ...
```

#4 - 05/19/2016 09:09 PM - Semen Yesylevskyy

Surely, I can do this. I can also use python scripts to generate mdp's and so on. However, each non-trivial preparation step increases your chances of making a mistake. Why should one use a workaround if it is easy to fix in the code? This is just a matter of input parsing, no rocket science.

#5 - 05/19/2016 09:25 PM - Chris Neale

The reason I believe one should use a workaround when it is easy to fix the code is that what you propose as a fix don't seem like a fix to me.

I'm only suggesting some further thought since the change that is an improvement for you (saves you some time and hassle to be sure) might cost a novice user a month of running something that they intended to be a pull-code run whereas they simply forgot to turn on pull=yes and then they silently got no pulling at all. Something similar happened to me once when I added some type of restraints in the .top file (possible dihedral restraints) and (assuming that this is the case) I didn't notice that I had to also turn on dihedral restraints with dihre = yes in the mdp file so I got a run without restraints but no warning at all.

#6 - 05/20/2016 02:35 PM - Erik Lindahl

- *Tracker changed from Bug to Feature*

- *Status changed from New to Rejected*

Agree this is not ideal; the reason for this behavior is that the pull options are read and treated in the separate pull source files, rather than the rest of the parameters that are processed in readir.cpp.

We won't be able to fix that for the 2016 release, but my reason for changing it into a feature is that we're just at a GROMACS workshop in Göttingen, and we've pretty much agreed how we should push ahead and move these formats to JSON instead. Then we will try to move all module-specific options into a JSON schema defined in each module, and that get rids of the problem.

#7 - 05/20/2016 02:37 PM - Erik Lindahl

- *Status changed from Rejected to Accepted*

Sorry, status should be "new feature", not rejected.