

GROMACS - Feature #2545

Should grompp fix periodicity of input files?

06/05/2018 08:43 PM - David van der Spoel

Status:	New	
Priority:	Normal	
Assignee:	David van der Spoel	
Category:	preprocessing (pdb2gmx,grompp)	
Target version:	future	
Difficulty:	simple	
Description		
If an input coordinate file is broken due to PBC, should grompp fix it? The issue is complicated if there are periodic molecules, in which case grompp should not touch the input file.		
Having tpr files with broken molecules may mess with analysis programs.		
Maybe a warning in grompp is sufficient?		
Related issues:		
Related to GROMACS - Bug #2544: gmx rmsf does not fix periodicity in referenc...		New

History

#1 - 06/05/2018 08:44 PM - David van der Spoel

Related to [#2544](#)

#2 - 06/05/2018 10:37 PM - Erik Lindahl

If we think about this from the user's point-of-view, he or she will typically just be asking "fit frame X on the reference".

Thinking of the analysis tools: Can we imagine any case where we should **NOT** make both the reference and frame whole before doing that?

Thinking of grompp/mdrun: To me, it seems reasonable that we by default make the input whole simply to avoid having to worry about it, unless the molecule is periodic.

Friend-of-order will then of course ask how we should fit periodic molecules - we probably want to decide that too while we're at it :-)

#3 - 06/05/2018 11:25 PM - David van der Spoel

OK, time to upgrade pbcutil/rmpbc to a c++ class?

Fixing the grompp (.tpr) and mdrun (.gro) output is relatively easy of course.

#4 - 06/06/2018 10:20 PM - Berk Hess

I see no reason why grompp should not do this, We should not put molecules in the box as well, unless they are broken to start with.

But any tool can also make the reference structure whole, but that should rather happen in the analysis framework, so it's only in one place and always happens.

#5 - 06/13/2018 12:32 PM - Mark Abraham

- Related to Bug #2544: gmx rmsf does not fix periodicity in reference structure added

#6 - 06/13/2018 06:41 PM - Mark Abraham

I have long suspected that some problems users have with analysis tasks stem from using confout.gro with broken molecules after preparation phases as input to grompp for production, and then using that production tpr file for analysis.

I am happy for grompp to make molecules whole, and I wonder if we can use the graph code to discover that a molecule is periodic so that grompp ignores those molecules while fixing any others? We ought not continue to produce second-class tpr files.

#7 - 06/13/2018 06:51 PM - Erik Lindahl

... But that still leaves the danger if somebody tries to fit a periodic molecule we did not make whole (not to mention old TPRs), or if I suddenly call a fitting code with some step-derived data.

It seems to me this is another example of code where we need to be much pickier with the input. A safe solution would be to require some sort of

connectivity/topology when doing PDC-sensitive ops, and then always check both the reference and data to ensure they have similar periodicity - and fail if not.

#8 - 06/13/2018 06:59 PM - David van der Spoel

We could write a "check_for_inconsistent_shifts" function, in fact the code is there, use that in grompp to check input structures.

However to fix this properly we would need a flag "periodic_molecule" per molecule(type), not just system wide at the mdp level.

To give another example, we have done simulations with a periodic DNA molecule (infinite chain), binding to proteins. It was quite a hassle to compute the RMSD of the proteins due to all the warnings. Computing distances between DNA and protein was even more tedious.

#9 - 06/13/2018 08:54 PM - Mark Abraham

Ah right, a periodic molecule presumably can be broken, too, and so should be made whole.

In some cases for some tools, we do insist on a .tpr file, but I assume we don't require it everywhere that we should.

We could bump the .tpr version for GROMACS 2019 such that its grompp makes molecules whole, and insist on such a tpr for tools that do fitting. Trying to magically do good behaviour with old .tpr will give people different results than they used to get, and if they don't understand why they do, that will lead to bug reports about what is (should be) correct behaviour...

#10 - 08/13/2018 02:06 PM - Szilárd Páll

- *Target version changed from 2018.3 to 2019*

We can't target a patch release with this feature, can we? I'll switch to 2019.

#11 - 10/03/2018 11:32 PM - Mark Abraham

- *Target version changed from 2019 to future*

I assume nobody can make this happen in time.