# GROMACS - Bug #1583

## gmx msd with mol flag requires excessive memory

08/25/2014 02:25 PM - Michael Brunsteiner

| | | | |
|---|---|---|---|
| **Status:** | New | | |
| **Priority:** | Normal | | |
| **Assignee:** | | | |
| **Category:** | analysis tools | | |
| **Target version:** | | | |
| **Affected version - extra info:** | | **Difficulty:** | uncategorized |
| **Affected version:** | 5.0 | | |

**Description**

I find that calculation of diffusion coeffs using g_msd together with the mol flag
on my desktop (64bit ubuntu 12.04) renders my computer in-operable as after a short time the
job uses ALL the memory, and eventually i have to kill the process.
also "gmx msd" jobs running on our rockscluster, e.g., with a trajectory containing 150 molecules
each with about 50 atoms are killed due to excessive memory usage after reading
approx 5000 frames ... (thus this seems to happen indepentedly of compiler version, libc, etc...)

```
echo "api" | gmx msd -f md.trr -s md.tpr -n md.ndx -o md-msd.xvg -mol md-mol-msd.xvg
...
Reading frame 5000 time 10000.000
/opt/gridengine/default/spool/node-0-2/job_scripts/63097: line 17:
26842 26843 Killed ...
```

this happens, both, with 4.6.5 and 5.0.
it seems that this issue has been reported a while ago (+3y) here:
http://redmine.gromacs.org/issues/774

it says there: "[...] The memory issue is unrelated and you should open a
separate issue for it [...]" ... but apparently this has never happened.
any ideas?

cheers,
michael

## History

**#1 - 08/25/2014 03:59 PM - Mark Abraham**

It certainly allocates O(nframes * nmols) memory. Fixing that is probably not very easy, but you can work around the issue by making index groups with single molecules. Inconvenient, I admit.

**#2 - 08/25/2014 04:14 PM - Michael Brunsteiner**

> It certainly allocates O(nframes * nmols) memory.

fine, but what i don't understand is this ... if i don't use the mol
flag the diffusion is calculated for all atoms in the provided group
which i'd expect to require O(nframes * natoms) memory, i.e. the problem
should become worse, but in this case the program works ...

> making index groups with single molecules.

you mean calculating the diffusion for each molecule separately
followed by averaging? that would take ages ... e.g. if i have 500 molecules
i'd need to read the entire trajectory 500 times ...

i guess a work around could be to 1) convert the original trajectoy

replacing all atomic coords of each molecule by the COM of the molecule,
and then 2) use gmx msd without the mol flag ... but there seems to no tool for
doing the first step ... so i'll try to write my own program for that
question is: is what is found in /usr/local/gromacs/share/gromacs/template/
known to work with gmx 5.0??

cheers
michael

**#3 - 08/26/2014 10:21 AM - Mark Abraham**

Michael Brunsteiner wrote:

> > It certainly allocates O(nframes * nmols) memory.
>
> fine, but what i don't understand is this ... if i don't use the mol
> flag the diffusion is calculated for all atoms in the provided group
> which i'd expect to require  O(nframes * natoms) memory, i.e. the problem
> should become worse, but in this case the program works ...

O(nframes*natoms) is the whole trajectory, which doesn't happen. Your second run is using a different code path, so the question is different.

> > making index groups with single molecules.
>
> you mean calculating the diffusion for each molecule separately
> followed by averaging? that would take ages ... e.g. if i have 500 molecules
> i'd need to read the entire trajectory 500 times ...
>
> i guess a work around could be to 1) convert the original trajectoy
> replacing all atomic coords of each molecule by the COM of the molecule,
> and then 2) use gmx msd without the mol flag ... but there seems to no tool for
> doing the first step ... so i'll try to write my own program for that

gmx traj does 1. Maybe then gmx analyze for 2?

> question is: is what is found in /usr/local/gromacs/share/gromacs/template/
> known to work with gmx 5.0??

The template code itself works, but the installed build system gear in 5.0 has issues that are being worked on, possibly resolved by now. Best bet includes configuring GROMACS originally with -DGMX_CXX11=off

> cheers
> michael

**#4 - 03/31/2015 01:17 PM - awen thomas**

Hi Michael,
Use the following code for solving this problem

> g_msd  -n POPC.ndx  -lateral z -o POPC_msd.xvg -mol POPC_diff.xvg
>
> Trajectory has 10000 frames and the system it was ran on is Fedora Red Hat 5.4.
> Indeed my network administrator was very unhappy about comsumed memory.

I'd guess that using only part of your trajectory at one time (g_msd -b
-e, or pre-condition with trjconv) will lead to less memory usage -
although the implementation of this code should not require 600GB unless
you actually have many millions of atoms.

Hope this code will work.