GROMACS - Bug #1035

g_cluster issues

11/15/2012 03:50 PM - Sergei Khruschev

Status:	Closed		
Priority:	Normal		
Assignee:	David van der Spoel		
Category:	analysis tools		
Target version:			
Affected version - extra info:		**Difficulty:**	uncategorized
Affected version:			

Description

Digging into g_cluster code showed something that seem a bit strange for me: when asking for the first index group, it claims this group will be used "for least squares fit and RMSD calculation", however it's actually used only for fitting and RMSD calculation is performed over the whole superposition of both selected index groups (see gmx_cluster.c line 1299). This line is supposed to be fixed to sth like:

rmsd = rmsdev_ind(ifsize,fitidx,mass,xx[i2],x1);

however similar fix is required for RMS distance deviation matrix calculation code.
And there is performance issue: if -nofit option is not specified, there is no need to copy each structure into a new array (lines 1295-1296). Combining this with previous issue, the following code is suggested:

```
if (bFit) {
        for(i=0; i&lt;isize; i++)
          copy_rvec(xx[i1][i],x1[i]);
        do_fit(isize,mass,xx[i2],x1);
        rmsd = rmsdev_ind(ifsize,fitidx,mass,xx[i2],x1);
      } else {
        rmsd = rmsdev_ind(ifsize,fitidx,mass,xx[i2],xx[i1]);
      }
```

And finally 'diagonalization' method issue: using this method without externally calculated RMSD matrix causes RMSD matrix to consist of NaN-s due to the fact that atom masses are not read by read_tps_conf (at line 1191 as bAnalyze is 0) but are used in RMSD calculation.

Associated revisions

Revision a973d598 - 11/26/2012 12:17 PM - David van der Spoel

Fixes #1035 NaN in g_cluster output.

Now always initializes the masses of the atoms to prevent division by zero downstream.

Change-Id: I1b38ccc7982d4340ed068535f7b7dd8e75e1a4c4

History

#1 - 11/19/2012 02:11 PM - Sergei Khruschev

Hmm... Sorry, it's not really a bug but just a performance issuue as "mass" array contains zeroes for atoms that are not in the fit/rmsd group.

#2 - 11/26/2012 10:50 AM - David van der Spoel

Hm just checking, but do you indeed get NaN in your matrix calculations or not?

#3 - 11/26/2012 11:22 AM - Sergei Khruschev

Yes, I do. Matrix consists of 'NaN' if '-method diagonalization' is specified... This issue has been mentioned in list several times and the answer always was "there is sth wrong with third party libs", but as I understand it is caused by call to

read_tps_conf(ftp2fn(efTPS,NFILE,fnm),buf,&top,&ePBC,&xtps,NULL,box,bAnalyze);

with bAnalyze=0 as

bAnalyze = (method m_linkage || method m_jarvis_patrick || method == m_gromos);

However I couldn't get what RMSD matrix diagonalization do with clustering... Is there any paper on this?

#4 - 11/26/2012 12:00 PM - David van der Spoel

No paper as far as I know. This may in fact be some old trial code of mine.
Do you use a tpr file as reference for the -matrix diagonalization issue? Could you try?
The difference due to bAnalyze is namely only whether or not the masses are initialized in case the input file is not a tpr file. However, in case of a tpr file the masses should always be initialized.

#5 - 11/26/2012 12:12 PM - Sergei Khruschev

That should be the cause - I use PDB file as reference, and it works fine for other methods... What's the idea of using bAnalyze as read_tps_conf() parameter? Passing 1 as last param to read_tps_conf() results in matrix being calculated. Is there any special reason for passing 0 here?

#6 - 11/26/2012 12:18 PM - David van der Spoel

No. I fixed it for 4.5.6 and upward, thanks for reporting.

#7 - 11/26/2012 06:05 PM - David van der Spoel

- Status changed from New to Closed