GROMACS - Feature #2132

Intermediate code for xvg handling

03/06/2017 04:02 PM - David van der Spoel

Status:	New
Priority:	Normal
Assignee:	
Category:	analysis tools
Target version:	future
Difficulty:	hard

Description

The transition from old analysis tools to new ones seems to be a big hurdle as no existing tools have been ported. Could the transition be eased by C++ifying parts of the code? An attempt was made to do this for the xvg handling code, however, in order to make it useful all the tools using it would have to be rewritten to some extent. Would it be better to cease any development (except bugfixes) to existing tools and require development of a new tool for every proposed change?

Associated revisions

Revision f4dd51d8 - 03/12/2017 10:11 AM - David van der Spoel

Added tests for gmx clustsize.

Some bone headed stuff to allow development of a new
version later. Unfortunately this now touches code
outside the analysis tool due to memory leaks that have to be
fixed.

Part of #2132

Change-Id: I32558e3a7cd7448954cf093208c15ce7014942ff

History

#1 - 03/06/2017 10:11 PM - Gerrit Code Review Bot

Gerrit received a related patchset '13' for Issue #2132.
Uploader: David van der Spoel (davidvanderspoel@gmail.com)
Change-Id: gromacs~master~I96285b2ca92d3e0441be050b82c7516fb828e619
Gerrit URL: https://gerrit.gromacs.org/6498

#2 - 03/07/2017 07:40 PM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue #2132.
Uploader: David van der Spoel (davidvanderspoel@gmail.com)
Change-Id: gromacs~master~I32558e3a7cd7448954cf093208c15ce7014942ff
Gerrit URL: https://gerrit.gromacs.org/6501

#3 - 03/08/2017 05:12 PM - Mark Abraham

In practice, I need to treat any proposed change to old-school tools functionality as low priority. I have too many open projects and too many calls on my time.

However, work that adds tests and ports an old tool to the new framework is of much higher interest, and that's the only pathway I can recommend for people wanting to contribute a new tool / feature.

One possible path to progress is to work out how to use the analysis framework's machinery for building the data sets assembled in an old-school tool (I don't know how feasible this is). That would provide for a dev workflow that produces commit series that look something like

- add test coverage (probably fixing a host of minor leaky things)
- break old tools into distinct parts (now that the wrapper binary implementation means that there is much lower pressure against proliferation of module names)
- build data sets with new machinery (above tests show that this is working fine)
- port the logic of the old-school test runner into the analysis framework approach

I have some work somewhere splitting gmx density into two parts and upgrading one of them, but it hasn't got to the point where I was ready to propose it to the world.