GROMACS - Task #1323

determine future of existing tools for

08/21/2013 11:42 PM - Mark Abraham

Status:	New
Priority:	Normal
Assignee:	David van der Spoel
Category:	analysis tools
Target version:	
Difficulty:	uncategorized

Description

We have about 95 tools. Those that are pre-processing and general support code will be retained, perhaps with updates as APIs emerge:

genrestr - remove (unless selection code + existing output routines will be painless)
pdb2gmx - retain in current form
grompp - retain, but do any necessary clean up so that a one-shot call to mdrun is also possible
mdrun - retain, all bets are still off!
tpbconv - retain in current form
trjconv - retain but make explicit splits of its functionality (one tool, one function principle), and add support for dynamic selections
g_x2top - retain in current form
genbox - retain in current form
genconf - retain in current form
genion - retain in current form
g_tune_pme - retain in current form
g_pme_error - retain (Xavier Periole has plans to update it)
xpm2ps - retain only if we still write xpm instead of some matrix format that actually has numbers people can actually use

That leaves about 85. We are not going to have the manpower to port more than a core set of them to the new analysis framework. For outright removal, I nominate:

g_protonate - remove
g_enemat - remove
g_sigeps - remove
g_morph - remove
g_anadock - remove, this is not even a GROMACS tool
g_dyndom - remove, this is not even a GROMACS tool
g_dih - already went before 4.6

In particular, these tools should probably change form quite a bit:
editconf - keep, probably implement core with analysis framework, niche functionality should disappear
g_select - generally superseded by support in new framework, but need to retain the ability to write an index-like file
make_ndx - retain, or implement as static selection from new framework
mk_angndx - retain, or implement as static selection from new framework

A bunch of the analyses should be fairly straightforward to implement as
1) make (static or dynamic) selection
2) loop over trajectory frames, filter those that don't suit any applicable criteria
3) apply selection, do any computation in a module, write output

There are current modules for computing a displacement, histogram, average & standard deviation; clearly there's a few more we'd like to have. Adding some kind of useful output matrix format is something to which I will contribute. There's a partial implementation of g_dist and g_angle in master branch now. Doing much of anything more needs us (me) to pull out some thumb and do some work so we can agree on the future of rvec.

The remaining ~70 analysis tools are not going to port themselves. On form and history, I expect that most of them will just sit there because they were written for one-time use for some publication and nobody cares any more. Thanks to just part of Teemu's heroic volunteer work, master branch supports "gmx g_rama -whatever" and I expect that this is how many of the tools will remain until someone wants features like dynamic selections and/or any parallelism support that emerges. If having to do a two-pass "trjconv for dynamic selection then call analysis tool" annoys people enough then we'll get some contributions of code, but the magic C++ code fairy has already moved mountains for us...

Fringe functionality of any tool that is ported will likely be omitted - particularly if it is readily/better done with some kind of external script. Adding it back later is much better than guessing in advance what people actually want.

Which of our existing tools would most benefit from dynamic selections, so that we can best focus on something useful to deliver for 5.0?

Even if you can't contribute much for lack of time or C++ skills, you can certainly help by offering characteristic input and output. One of the big problems is that nobody knows what combinations of tools' options actually work. Input+output+somebody's belief in existing correctness is what will be needed for porting any analysis tool for which the correctness of the implementation cannot be assessed with confidence by inspection of the code!

Related issues:		
Related to GROMACS - Task #665: Port existing trajectory analysis tools to us...	**New**	**03/27/2012**
Related to GROMACS - Feature #921: Default index groups and selections	**Blocked, need info**	
Related to GROMACS - Task #1971: Removing buggy features vs. keeping workflows	**New**	
Related to GROMACS - Bug #2037: trjconv produces gro files unreadable by VMD	**Closed**	

Associated revisions

Revision c5e057f3 - 12/04/2018 11:17 AM - David van der Spoel

Removed gmx anadock

Part of #1323

Change-Id: I77c04e08a8245db38530e9aa4113447cc02672a8

Revision 61adbf69 - 12/04/2018 11:26 AM - David van der Spoel

Removed gmx dyndom

Part of #1323

Change-Id: I07f5cfed364e13fbd6a8b6e0fc94b4b15ef67589

Revision 2ea2d842 - 12/04/2018 11:53 AM - David van der Spoel

Removed gmx morph

Part of #1323

Change-Id: Ife56a50e61e4859fa20a39a1be59e828d511fd09

History

#1 - 08/22/2013 08:16 AM - David van der Spoel

Thanks for bringing this up Mark, and for keeping us on our toes.

I have just introduced extensive changes in genrestr which do not make sense to put elsewhere, this is in a branch which I hope to merge into master this fall. Therefore I would like to keep it.

g_x2top will be replaced by a new tool that generates GAFF topologies (is working already but needs more testing).

Apart from that we will in my group work on porting analysis tools to the new framework, and develop new ones as well. But this is as usual prioritized by the science. g_hbond is on top of the list for porting as it will profit from parallellization.

#2 - 08/22/2013 08:39 AM - Erik Lindahl

Hi,

I would still suggest aggressively removing most tools that are not well-modularized, well-documented, small and clean for now. If these tools are then cleaned up we can reintroduce them. Another alternative could be to keep some of these tools for now if they are really important, but not accept any extensions or new options in them until they have been cleaned up.

On the one all of us need to prioritize science, but if that means the associated tool is buggy, not documented or otherwise difficult to maintain, that also means they might have to be maintained outside the main Gromacs repository, for instance as a "contributed" tool (even if it comes from a developer group). I think g_bhond is a good example of a tool where we should think of removing some options to make a leaner and cleaner tool before it is ported to the new framework. Of course, this does not rule out an alternative highly advanced version like g_bhond_advanced that's available as a contributed tool.

#3 - 08/22/2013 05:49 PM - Teemu Murtola

First quick set of comments, more to come...

Mark Abraham wrote:

> pdb2gmx - retain in current form
> g_protonate - remove

g_protonate is actually one of the few tools that I've used. The functionality is probably anyways required in pdb2gmx, so having a tool that does just that does make sense. I think there has been talk about splitting pdb2gmx into smaller pieces, and g_protonate is one such relatively self-contained piece.

> In particular, these tools should probably change form quite a bit:
> g_select - generally superseded by support in new framework, but need to retain the ability to write an index-like file

This is one course of action. Currently, g_select computes some properties of the selections themselves such as the size of the selections. This provides some very useful functionality, and may not find an easy-to-find home elsewhere. Some functionality definitely can move into a more specialized tool.

> make_ndx - retain, or implement as static selection from new framework

This provides more or less identical functionality to g_select -on. The features that g_select currently lacks:

- Ability to generate default groups ([#921](#))
- Ability to split groups into atoms/residues/whatever

> There are current modules for computing a displacement, histogram, average & standard deviation; clearly there's a few more we'd like to have.

The displacement module is not in a good shape, and would really need a rewrite ([#909](#)), but otherwise, the mentioned modules should provide basic functionality. Calculation of autocorrelation functions is probably a good candidate for a next general-purpose module.

> There's a partial implementation of g_dist and g_angle in master branch now.

I just extended the implementation, so that the new distance module should cover all functionality of g_bond and g_dist. The new angle module actually provides a greatly extended version of g_sgangle, and as a side effect, also part of g_angle functionality.

> Which of our existing tools would most benefit from dynamic selections, so that we can best focus on something useful to deliver for 5.0?

At some point of time, I've had a selection-enabled version of the following tools (actually used for research), in addition to those currently in master:

- g_msd
- g_sas
- g_sorient (could be better implemented as a postprocessing script that combines the results from the new distance and angle modules)
- g_rdf

#4 - 08/25/2013 06:32 AM - Teemu Murtola

As a side note, it was noted in https://gerrit.gromacs.org/#/c/2519/ that ngmx and g_xrama (or GMX_X11=ON in general) isn't currently verified by Jenkins. If we want to keep those, then at minimum, we should have at least one Jenkins configuration build them to avoid breaking compilation...

#5 - 08/30/2013 09:35 PM - Teemu Murtola

Mark Abraham wrote:

> trjconv - retain but make explicit splits of its functionality (one tool, one function principle), and add support for dynamic selections

That is one possibility, but in general, I think it would be nice to make it possible to invoke trjconv as little as possible. In particular with very big trajectories, having to keep ten different post-processed or intermediate copies of the trajectory to be able to run different tools on it is not very nice. And processing those takes a large fraction of the analysis time, since I/O is the limiting factor in many analyses. Some options could be included in the tool framework (currently, it provides an option to make molecules whole). For trjconv, modularizing it is certainly worth the effort, but it would be best if one would not need to run the tool more than once to get the desired outcome.

Support for dynamic selections is probably nice, but see below a comment about somewhat limited usability.

> The remaining ~70 analysis tools are not going to port themselves. On form and history, I expect that most of them will just sit there because they were written for one-time use for some publication and nobody cares any more. Thanks to just part of Teemu's heroic volunteer work, master branch supports "gmx g_rama -whatever" and I expect that this is how many of the tools will remain until someone wants features like dynamic selections and/or any parallelism support that emerges.

This is probably the best course of action. Erik's comment was also along these lines. Removing tools that work (or are very likely to work; without

tests there aren't any guarantees) may be counterproductive, as I've heard comments that some people actually choose Gromacs because of the analysis tools.

If having to do a two-pass "trjconv for dynamic selection then call analysis tool" annoys people enough then we'll get some contributions of code, but the magic C++ code fairy has already moved mountains for us...

Such a two-pass analysis is likely not going to work with existing tools except in some very special cases. I know that on the mailing list, people seem to request the ability to write trajectories with dynamic selections now and then, but existing tools will probably choke on a changing number of atoms (or even worse, changing topology) between frames.

#6 - 06/19/2014 01:19 PM - David van der Spoel

- Target version changed from 5.0 to 5.x

#7 - 05/27/2016 12:31 AM - Mark Abraham

- Related to Task #1971: Removing buggy features vs. keeping workflows added

#8 - 07/11/2016 08:16 PM - Mark Abraham

- Target version deleted (5.x)

#9 - 08/23/2016 03:18 PM - Mark Abraham

Some discussion of possible future directions of trjconv took place at #2037

#10 - 08/23/2016 03:18 PM - Mark Abraham

- Related to Bug #2037: trjconv produces gro files unreadable by VMD added

#11 - 03/06/2018 11:29 PM - Mark Abraham

- Subject changed from determine future of existing tools for 5.0 to determine future of existing tools for

#12 - 05/22/2018 05:22 PM - Mark Abraham

Case in point (#2511): gmx editconf -mead used to write a non-conformant PQR file. If we would keep this functionality in the longer term, it should really be e.g. a single function that implements gmx convert -f topol.tpr -o mead.pqr.

#13 - 12/04/2018 11:12 AM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue #1323.
Uploader: David van der Spoel (spoel@xray.bmc.uu.se)
Change-Id: gromacs~master~I77c04e08a8245db38530e9aa4113447cc02672a8
Gerrit URL: https://gerrit.gromacs.org/8775

#14 - 12/04/2018 11:16 AM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue #1323.
Uploader: David van der Spoel (spoel@xray.bmc.uu.se)
Change-Id: gromacs~master~I07f5cfed364e13fbd6a8b6e0fc94b4b15ef67589
Gerrit URL: https://gerrit.gromacs.org/8776

#15 - 12/04/2018 11:33 AM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue #1323.
Uploader: David van der Spoel (spoel@xray.bmc.uu.se)
Change-Id: gromacs~master~Ife56a50e61e4859fa20a39a1be59e828d511fd09
Gerrit URL: https://gerrit.gromacs.org/8777

#16 - 12/04/2018 11:53 AM - David van der Spoel

I would suggest to keep sigeps as it is small and sometimes useful. However when the t_pargs are removed this will have to be updated as well.