Gromacs - Task #708

Refactor options module to support adjusting option settings after creation

02/19/2011 12:49 AM - Teemu Murtola

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>Teemu Murtola</td>
</tr>
<tr>
<td>Category:</td>
<td>core library</td>
</tr>
<tr>
<td>Target version:</td>
<td>5.0</td>
</tr>
</tbody>
</table>

Description

In analysis tools, the number of selections (or the flags for the required selections) required may in some cases be known only after other options have been parsed. To make interactive selection input work well in such cases, these changed settings should be updated in the request for interactive input. Implementing all this access to go through SelectionOptionStorage would make it cleanest: all user input checking could be done there, both when the selections have already been provided on the command line, or when the selections will be provided interactively later. However, the storage objects are not designed to be exposed to the callers; the mutable parts should either be refactored to a separate class, or a wrapper class implemented.

This feature could either be implemented just for selection options (keeps the core options module a bit simpler), or it could be made more general by making a similar refactoring in AbstractOptionStorage. Even in the first case, AbstractOptionStorage still needs to be adapted to support changing the limits for the number of values at any point.

Related issues:

- Related to Gromacs - Task #839: Improve OptionsIterator/OptionsGlobalProperti... 
  Closed 11/14/2011
- Blocks Gromacs - Task #665: Port existing trajectory analysis tools to use th... 
  New 03/27/2012

Associated revisions

Revision a37cc2e5 - 02/23/2011 05:25 PM - Teemu Murtola

Combined and renamed options-related files.

Merged *optionstorage.cpp files into corresponding *option.cpp files, because there was no real reason to have them separate, and this will make it easier to find all code related a certain option. In particular with issue #708, which will introduce classes whose definition should go to the *option.h header file, but whose implementation would be more natural in *optionstorage.cpp.

Refactoring in preparation for IssueID #708.

Revision 9aa9d065 - 03/03/2011 07:44 PM - Teemu Murtola

Allow selection option adjustment after creation.

Added functionality that allows one to adjust selection options (e.g., the number of selections required) based on values of other options.

IssueID #708

Revision 6207afb9 - 03/03/2011 08:39 PM - Teemu Murtola

Bug fixes to previous commit.
The angle tool did not work properly due to a few bugs in the previous commit. Changed the analysis module interface slightly to provide an error reporter to where it's needed. Also fixed some error messages.

IssueID #708

**History**

**#1 - 02/23/2011 06:20 AM - Teemu Murtola**
- Status changed from New to In Progress
- Assignee set to Teemu Murtola

**#2 - 03/03/2011 07:50 PM - Teemu Murtola**
- Status changed from In Progress to Closed

Took the simplest way, and implemented a wrapper class that provides an interface that allows adjusting options after creation. The wrapper simply passes the calls to the storage object for actual processing to keep the logic in one place. This design keeps the core options module as simple as possible, and keeps the storage object well isolated from users.

Only implemented this for SelectionOption, but the more general parts of the implemented SelectionOptionAdjuster should be easy to refactor out to allow them to be reused for other options if the need ever arises.

**#3 - 02/16/2014 08:58 PM - Teemu Murtola**
- Related to Task #839: Improve OptionsIterator/OptionsGlobalProperties interface added

**#4 - 02/16/2014 08:59 PM - Teemu Murtola**
- Project changed from Next-generation analysis tools to Gromacs
- Category set to core library
- Target version set to 5.0