

GROMACS - Task #938

Regular expressions in selections

04/30/2012 06:43 PM - Teemu Murtola

| | |
|---|---------------|
| Status: | Closed |
| Priority: | Normal |
| Assignee: | Teemu Murtola |
| Category: | selections |
| Target version: | 5.0 |
| Difficulty: | uncategorized |
| Description | |
| <p>Currently, string keywords in selections (such as "resname") can be matched against regular expressions using POSIX regex if that is available. If it cannot be found during configuration, this support is silently dropped, and this changes the behavior of selections. The selection code also tries to guess whether the input string is a regex or a simple pattern using ? and * wildcards only, and does the matching based on this guess.</p> <p>It would be better that the same syntax for selections would always work the same way, and give an error if regexp support is required but not available. This would require a mechanism to specify whether something needs to be matched using regexps or not. There are (at least) two alternatives:</p> <ol style="list-style-type: none">1. Use different types of quotes like VMD does (single quoted strings are matched literally, doubly quoted as regexps, or something similar).2. Use special keyword(s) that switches/forces the matching mode, e.g., resname regexp "R[AB]", resname nowildcards "R**". <p>Neither is particularly difficult to implement (the second can be done without touching the actual selection parser code, the first requires some simple code to pass the type of quotes as part of strings), so opinions would be welcome.</p> | |

Associated revisions

Revision a9beaee9 - 08/28/2012 05:45 AM - Teemu Murtola

Add syntax to force selection string matching mode.

The selection syntax for string keyword matching no longer depends on whether regular expression support is available. Instead, an error is now given if the string looks like a regexp (the logic for the deduction is not changed), but regexp support is not available. Added syntax to force the string matching to use either literal, wildcard, or regexp matching.

This change allows removing a few more direct prints to stderr (related to #655).

Closes #938.

Change-Id: I7b998050c8b00b5f1229ed23a0a15685c514010f

History

#1 - 08/03/2012 01:47 PM - Roland Schulz

I like option 2 better. I think using two different quotes is not very intuitive. I would suggest "resname ~ "R[AB]" if it isn't too difficult. But using "regexp" or "like" instead of "~" is OK too. The only advantage of "~" is that it is shorter.

#2 - 08/03/2012 09:33 PM - Teemu Murtola

Thanks for the idea. I think that resname ~ "R[AB]" should be easy enough to implement, and a non-alphanumeric character makes it syntactically a bit less ambiguous. Syntax for forcing plain string matching could then perhaps be resname = "R*" "RA", although there's some possibility for confusion here (it requires less code to use a single-char symbol than ==, but that one could be used as well)...

There are now four possible cases that should be somehow distinguished:

- Plain string matching.
- Regexp matching.
- Wildcard matching.
- Automatic detection between regexp and wildcard matching.

The above syntax with either ~ or = to force the first two would leave the syntax resname "str" for either wildcard matching, or for automatic detection. Using it for automatic detection would leave current behavior unchanged, but then there would be no way of forcing wildcard matching. Not sure though if that would be a big problem. Adding yet another symbol (like * or ?) for the wildcard match could also be a bit confusing...

#3 - 08/05/2012 07:16 AM - Teemu Murtola

- Assignee set to Teemu Murtola

Will implement this after I have finished doing modifications to the parser for [#655](#) (which in turn requires [#985](#)).

#4 - 08/31/2012 06:24 AM - Teemu Murtola

- Status changed from New to Closed

#5 - 02/08/2014 06:57 AM - Teemu Murtola

- Project changed from Next-generation analysis tools to GROMACS

- Category set to selections