GROMACS - Bug #1007

g_select deals poorly with 4-character atom names (generated by pdb2gmx)

09/20/2012 09:56 PM - Peter Kasson

Status:	Closed		
Priority:	Normal		
Assignee:	Teemu Murtola		
Category:	selections		
Target version:	4.5.6		
Affected version - extra info:	4.5-4.5.5	**Difficulty:**	uncategorized
Affected version:	4.5.5		

Description

Git-current release-4-5 doesn't seem to recognize atoms with 4-character names properly:

PDB snippet:

```
ATOM   3735  N   ILE A 279      44.270  67.580  54.020  1.00  0.00
ATOM   3736  H   ILE A 279      43.850  68.480  53.840  1.00  0.00
ATOM   3737  CA  ILE A 279      45.360  67.130  53.080  1.00  0.00
ATOM   3738  HA  ILE A 279      45.120  66.140  52.690  1.00  0.00
ATOM   3739  CB  ILE A 279      45.380  68.140  51.890  1.00  0.00
ATOM   3740  HB  ILE A 279      45.500  69.130  52.340  1.00  0.00
ATOM   3741  CG2 ILE A 279      46.580  67.960  50.990  1.00  0.00
ATOM   3742 1HG2 ILE A 279      46.640  68.710  50.210  1.00  0.00
ATOM   3743 2HG2 ILE A 279      47.540  68.090  51.480  1.00  0.00
ATOM   3744 3HG2 ILE A 279      46.670  66.930  50.630  1.00  0.00
ATOM   3745  CG1 ILE A 279      44.120  67.930  50.980  1.00  0.00
ATOM   3746 1HG1 ILE A 279      43.280  67.620  51.600  1.00  0.00
ATOM   3747 2HG1 ILE A 279      44.420  67.140  50.290  1.00  0.00
ATOM   3748  CD  ILE A 279      43.650  69.190  50.230  1.00  0.00
ATOM   3749  HD1 ILE A 279      42.680  68.980  49.770  1.00  0.00
ATOM   3750  HD2 ILE A 279      43.670  70.000  50.950  1.00  0.00
ATOM   3751  HD3 ILE A 279      44.310  69.330  49.370  1.00  0.00
ATOM   3752  C   ILE A 279      46.700  66.810  53.880  1.00  0.00
ATOM   3753  O   ILE A 279      47.470  65.910  53.530  1.00  0.00
```

selection file snippet:
resid 0279 and chain A and name CG1;
resid 0279 and chain A and name 3HG2;
resid 0279 and chain A and name 1HG1;
resid 0279 and chain A and name 2HG2;

output ndx:
[resid_0279_and_chain_A_and_name_CG1]
3745
[resid_0279_and_chain_A_and_name_3HG2]

[resid_0279_and_chain_A_and_name_1HG1]

[resid_0279_and_chain_A_and_name_2HG2]

Associated revisions

Revision 79f2d06a - 09/24/2012 05:43 PM - Teemu Murtola

Add pdbname selection keyword.

As a supporting change, remove trailing space from t_pdbinfo.atomnm, as
the trailing whitespace does not seem to be used anywhere.
This makes it possible to use it easily in the selection code.

Fixes #1007; fix backported from Iac36bda8.

Also includes changes from 8bddac3 to make the backport easier.

Change-Id: Iac36bda8a84d0a6c131445e7f47ad91d7209fb10

Revision 79f2d06a - 09/24/2012 05:43 PM - Teemu Murtola

Add pdbname selection keyword.

As a supporting change, remove trailing space from t_pdbinfo.atomnm, as
the trailing whitespace does not seem to be used anywhere.
This makes it possible to use it easily in the selection code.

Fixes #1007; fix backported from Iac36bda8.

Also includes changes from 8bddac3 to make the backport easier.

Change-Id: Iac36bda8a84d0a6c131445e7f47ad91d7209fb10

Revision ccb2b415 - 09/24/2012 05:47 PM - Teemu Murtola

Add pdbname selection keyword.

As a supporting change, remove trailing space from t_pdbinfo.atomnm, as
the trailing whitespace does not seem to be used anywhere.
This makes it possible to use it easily in the selection code.

Fixes #1007 in master, will backport to 4.5 branch separately.

Change-Id: Iac36bda8a84d0a6c131445e7f47ad91d7209fb10

History

#1 - 09/21/2012 06:29 AM - Teemu Murtola

- Description updated

The issue is in the code that loads the PDB file: when a topology is loaded using read_stx_conf() (which is what g_select uses), 4-letter atom names starting with a digit are internally converted by moving the digit at the end. So 3HG2 becomes HG32, and this is what g_select sees. So there are two alternatives:

- Add special cases to the selection evaluation such that atom name matching works differently based on whether the input file is a PDB file or not. I personally think this is quite ugly, and difficult/impossible to get to work right with, e.g., regular expressions.
- Remove this translation from the I/O routines. Need to analyze the impact...

4-letter atom names work as expected with a gro file as an input.

#2 - 09/21/2012 12:30 PM - Teemu Murtola

I was in a bit of a hurry when writing the first comment, so here is a bit more thorough analysis of the situation. Current situation is that

- Whenever a PDB file is loaded into Gromacs, atom names of the form NABC (where N is a digit and the name is at least 4 chars) are transformed into ABCN for storage in t_atoms.atomname. The original atom name from the PDB file is stored into t_atoms.pdbinfo.atomnm, but this version is not trimmed of whitespace. It could be easier to work with if it was trimmed, but I don't know if the whitespace is significant to some current users of this information.
- Whenever a PDB file is written from Gromacs, a reverse transformation takes place, no matter what is the source of the original atom names (so taking a PDB input with atom name HG13 and writing that out produces 3HG1).
- Selections (and also, e.g., make_ndx) only use t_atoms.atomname for matching the atom names. So g_select and make_ndx both work the same.
- I assume that the reason for these transformations is somehow related to pdb2gmx, but I agree that the behavior of g_select (and also make_ndx) is not very intuitive.

Possible approaches to solve the problem:

1. Make name selections match t_atoms.pdbinfo.atomnm if it is present, otherwise use t_atoms.atomname. This resolves the unintuitive behavior in this change, but introduces another one: depending on the input format, the same selection may select different atoms. So if one just takes a tpr and a pdb file produced by the same invocation of mdrun (or a pdb file used as input to grompp, and the produced tpr file), the selection syntax will be different.
2. Make name selections match either t_atoms.pdbinfo.atomnm (if present) or t_atoms.atomname. Can also be quite unintuitive, as HG23 can then match both HG23 and 3HG2 in the input.
3. Make name selections always match against t_atoms.atomname, but also try applying the reverse PDB transformation. The unintuitive behavior from the second point is even stronger here.
4. Add a separate pdbname selection keyword that will match only t_atoms.pdbinfo.atomnm. It can either fall back to t_atoms.atomname or give an error for non-PDB input. In particular without the fallback, this keeps the selection syntax more predictable.

5. Try to keep the user-supplied atom names also when dealing with PDB files. There is probably some rationale (which I don't know) for this translation, so this may introduce other problems.
6. Keep the current behavior, but just document it. At least the current behavior is consistent between g_select and make_ndx, and also has the nice property that the same selection produces the same output for different input formats that were produced by Gromacs conversions.

Any of the alternatives except the fifth should be straightforward to implement.

#3 - 09/23/2012 11:07 PM - Peter Kasson

Hmm--I would think 4 + 6 would be good--if we document the behavior and then have a pdbname selection keyword to allow matching the pdb convention. We can leave the whitespace convention alone for the moment, although it might be nice to allow matches to non-whitespace pdbname if that is straightforward.

#4 - 09/24/2012 05:29 PM - Teemu Murtola

- Category set to analysis tools

- Status changed from New to In Progress

- Target version set to 4.5.6

- Affected version - extra info set to 4.5-4.5.5

#5 - 09/25/2012 04:44 PM - Teemu Murtola

- Status changed from In Progress to Closed

#6 - 07/17/2013 07:17 AM - Teemu Murtola

- Category changed from analysis tools to selections

- Affected version set to 4.5.5