GROMACS - Bug #3206

2 not-critical bugs in analyse tool

11/15/2019 02:29 PM - Boris Timofeev

<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></td>
</tr>
<tr>
<td>Category:</td>
<td>analysis tools</td>
</tr>
<tr>
<td>Target version:</td>
<td>2019.5</td>
</tr>
<tr>
<td>Affected version - extra info:</td>
<td></td>
</tr>
<tr>
<td>Affected version:</td>
<td>2019.4</td>
</tr>
<tr>
<td>Difficulty:</td>
<td>simple</td>
</tr>
</tbody>
</table>

Description

To solve the problem of building complete statistics on all angles and didehralss of modified DNA, (See issue # 3184), I had to write a small console program on C++ using the libgromacs API.

Two bugs was found and corrected:

1. The gmx_conect_addline parsing function of CONNECT fields in pdb files appears to contain errors in the format string, due to migrating from sprintf to std::string in 2019.4.

   Must be:

   ```c
   static void gmx_conect_addline(gmx_conect_t *con, char *line) {
      int         n, ai, aj;

      std::string form2  = "%s";
      std::string format = form2 + "%d";
      if (sscanf(line, format.c_str(), &ai) == 1) {
         do {
            form2 += "%s";
            format = form2 + "%d";
            n = sscanf(line, format.c_str(), &aj);
            if (n  1) {
               gmx_conect_add(con, ai - 1, aj - 1); /* to prevent duplicated records */
            }
         } while (n  1);
      }
   }
   ```

   In "git master" branch sprintf function is used and all is right.

   Gromacs never uses CONNECT records in pdb (source context search), but this is not a reason to contain knowingly inoperable code.

2. It is further revealed that the gmx angle for Std.deviation sometimes returns a NAN, i.e. use sqrt from a negative number.

   In a detailed review, it was revealed that:
   at first, it is mathematically incorrect to calculate the mean square deviation from the pre-calculated means, and second, float (32 bit) accuracy is lacking even if the standard deviation is properly calculated.

   I propose the following corrections in patch attached.

Associated revisions

Revision 7d142404 - 11/25/2019 08:10 PM - Test User

Fix duplicate CONNECT records in pdb files
When adding connect records in pdb files, it was not checked if records already existed and were thus duplicated in output. This patch fixes this behavior.

Fix originally provided by Boris Timofeev.

refs #3206

Change-Id: I5850f4b6279b37df07282a4416f433099d90c8

Revision 37516546 - 12/20/2019 01:29 PM - Test User

Remove problematic output of gmx angle

It could happen that the calculation of the standard deviation for angles caused a divide by zero error for empty populations. Because this standard deviation of per frame averages over all frames was anyhow meaningless, it has been removed.

Refs #3206

Change-Id: lcf541a8b3b6f02da9a22b55a5e8154b2d4f8

History

#1 - 11/15/2019 02:36 PM - Boris Timofeev
- File my_angle_analyze.patch added

Patch 2 must be used

#2 - 11/15/2019 03:53 PM - Boris Timofeev
- File pdb_analyse.cpp added

Source for "batch angle statistic", if it is interesting

#3 - 11/25/2019 04:56 PM - Christian Blau
The standard deviation that is calculated is also correct, though it is the standard deviation of the average angle per frame, instead of the overall standard deviation. I changed the output to reflect that better.

#4 - 11/27/2019 12:07 AM - Boris Timofeev
The average calculated from the frame averages is, at the same time, the average over the entire sample. You can’t say the same about the standard deviation. But the issue of loss of accuracy in calculating angles and the standard deviation remains open. Even if we use double to accumulate, the values obtained by calculating angles with float accuracy regularly result in a situation not possible to calculate the standard deviation.
And the using of float is simply unsuitable. Try to run next sample:

```c
int main()
{
    #define RAD2DEG (180.0/M_PI)
    typedef float MY_REAL;
    float a[] = { 119.634, 119.657, 119.634};
    MY_REAL sum_a = 0, sum_a2 = 0;
    int i;
    for (i = 0; i < sizeof(a) / sizeof(a[0]); ++i) {
        sum_a += a[i] * (180.0 / M_PI);
        sum_a2 += a[i] * a[i] * (180.0 / M_PI) * (180.0 / M_PI);
    }
    sum_a /= i;
    sum_a2 /= i;
    sum_a *= sum_a;
    assert(sum_a2 > sum_a);
}
```

assertion failed when MY_REAL => float, and not failed when MY_REAL => double

#5 - 11/27/2019 12:22 AM - Boris Timofeev
It's need also remove double "%" sign for version with std::string in pdbio.cpp:

```c
std::string form2 = "%%s";                !!!!!!!!!!!!! Here must be: std::string form2 = "%s";
```
std::string format = form2 + "%%d"; !!!!!!!! Here must be: std::string format = form2 + "%d";
if (sscanf(line, format.c_str(), &ai) == 1)
{
    do
    {
        form2 += "%*s";
        format = form2 + "%%d"; !!!!!!!! Here must be: format = form2 + "%d";

    #6 - 11/27/2019 12:27 AM - Boris Timofeev
    And another question about angles:
    We have two measured torsion angles: -179 deg and 179 deg.
    What is a mean value?
    I think, 180 degrees, not 0.

    #7 - 12/23/2019 01:12 PM - Paul Bauer
    - Status changed from New to Resolved

    I think this is done now

    #8 - 12/23/2019 01:53 PM - Paul Bauer
    - Status changed from Resolved to Closed

Files

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>my_angle_analyze.patch</td>
<td>2.83 KB</td>
<td>11/15/2019</td>
<td>Boris Timofeev</td>
</tr>
<tr>
<td>my_angle_analyze.patch</td>
<td>2.84 KB</td>
<td>11/15/2019</td>
<td>Boris Timofeev</td>
</tr>
<tr>
<td>pdb_analyse.cpp</td>
<td>9.67 KB</td>
<td>11/15/2019</td>
<td>Boris Timofeev</td>
</tr>
</tbody>
</table>