How to use GROMACS with S^2 and pairwise NOE distance restraints

Zoltán Gáspári

May 22, 2009

Contents

1 Introduction

This document describes the usage of S^2 and pairwise NOE distance restraints in GROMACS. These restraints are not included in the official version, rather programmed by myself (Z. Gáspári). Therefore, it is important that the version described here is **not official GROMACS** and should not be treated or considered as such in any way.

This document is practice-oriented, so the theoretical part and some implementation details will come only at the end.

Installation instructions can be found in the file named 'INSTALL' supplied with the package.

2 About ensemble simulations

S^2 restraints can only be used for multiple structures simlated simultaneously, so a bririef intoduction to ensemble calculations is in place. The official version of GROMACS (as of writing this is 3.3) is able to handle ensembles and use NOE and orientation restraints on them. GROMACS should be compiled with mpi support (most binary distributions are compiled without it). See the file 'INSTALL' supplied with the package for more information.

The magic command to run a simulation in ensemble mode is:

```
mpirun -np N mdrun -np N -multi -s topol.tpr -o traj.trr
```

Here the number N controls the number of replicas to simulate, note that it should be supplied twice, once for MPIRUN and once for MDRUN. The option *-multi* tells gromacs to simulate as many molecules as processes, this also causes mdrun to look for files *topol0.tpr, topol1.tpr ...* as inputs for each process. Note that for this type of runs you should run grompp as usual but separately for each replica to produce the N input (*tpr*) files[1] Note that the number N may be less or more than the number of processors in your system, mpirun can handle this for you (e.g. I usually run 8 replicas on a 2-processor machine). It is also important that in order to get meaningful results, your input *tpr* files should contain different starting structures and/or velocities (different starting velocities can be generated by in setting the *gen_vel* option to "yes" and the *gen_seed* to a different number for each replica in the *mdp* files).

3 Getting started

The directory 'test_vacuum' supplied with the package contains the files and scripts for a simple parallel simulation. Copy this directory to a place in your home diractory and try the scripts *setup.sh* and *run.sh*. If you understand these scripts, you can easily set up and run your own stuff. The scripts *rmsd.sh* and *getfinalstructures.sh* help you in the analysis of the runs.

4 Perl additions

There are a number of small PERL scripts to ease calculation setup and analysis (in the 'perl' directory of the package, copy/link them into a location in your PATH or modify your PATH to be able to use them). All programs are GROMACS-like and invoked with *-h* will give information on their usage. The programs are:

- MULTIGROMACS.PL This program helps you set up ensemble simlations as it runs a specified GROMACS command (specified with option *-C*) N times (option emph-N). Any '#' characters in the commands given will be substituted by the numbers 0..N-1 for each invocation. This program is especially useful for generating the input *topol0.tpr, topol1.tpr ...* files with grompp.

- MDP_RANDOMSEED.PL This program is used to set all the *"seed"* options in the input *mdp* file to a random number. If the otion is present in the input file, it will be overwritten, if not, it will be added.

- GROMACSRES2TOP.PL This script can be used to add restraints (NOE, S^2 and dihedral) to a GROMACS topology (*top*) file. Invoking with *-h* will give you an idea how to use it.

[1] This means that you should **not** use the -n switch for GROMPP, that should be used with distributed calculations where you would like to run a single simulation on multiple nodes. Of course you can run an ensemble calculation with each replica running as more than one process, then you should combine these two modes.

5 Input file special notes

5.1 Topology file

Pairwise distance restraints can be specified with the directive `hhdistance_restraints` in the same way as 'normal' distance restraints (see the official GROMACS manual for more):

```
[ hhdistance_restraints ]
; ai   aj  type index  type'  low   up1  up2   fac
    6   13    1    0     1   0.33  0.60 99.90  1.0 ;    1  HA:   1HD21
    6   14    1    1     1   0.33  0.60 99.90  1.0 ;    1  HA:   1HD22
    6   18    1    2     1   0.23  0.50 99.90  1.0 ;    1  HA:   2   H
    8   58    1    3     1   0.23  0.50 99.90  1.0 ;    1 HB1:   4   H
    9   58    1    3     1   0.23  0.50 99.90  1.0 ;    1 HB2:   4   H
    8   62    1    4     1   0.23  0.50 99.90  1.0 ;    1 HB1:   4  HB
    9   62    1    4     1   0.23  0.50 99.90  1.0 ;    1 HB2:   4  HB
```

(The prefix 'hh' refers to 'half-harmonic', meaning that a progress variable ρ is defined as described e.g. in Best and Vensrucolo, 2004 (for S^2 restraints)).

S^2 restraints can be specified with the directive `s2_restraints`:

```
[ s2_restraints ]
; ai   aj  type   S2   S2err  bondlen
   36   37  1    0.993   0.000  -1.000 ;    3   N:   3   H
   57   58  1    0.806   0.000  -1.000 ;    4   N:   4   H
   76   77  1    0.846   0.000  -1.000 ;    5   N:   5   H
```

The fields `ai` and `aj` contain the atoms in the bond (amide N-H in the example above), `type` shoule be one, the S^2 value and its error are put in the fields `S2` and `S2err`. The parameter `bondlen` may contain a positive value corresponding to the bond length to be used for S^2 parameter calculation or -1 meaning that the length is taken from the input structure.

5.2 mdp file

The relevant part of the `mdp` file for setting up pairwise distance and S^2 restraints looks something like:

```
; Half-harmonic distance restraints
; Distance restraints type: No, Simple,  Ensemble, Pairwise (=pairwise averaging in the ensemble)
hhdisre                  = pairwise
hhdisre_fc               = 10000
hhdisre-tau              = 0
hhdisre-exp              = 3
; Output frequency for pair distances to log file
nsthhdisreout            = 100

; S2 restraints: No,  Withfit, Withoutfit
s2re                     = Withfit
; Output frequency for S2 values to log file
nsts2reout               = 1000
```

```
; S2 force constant (denoted alpha in Best & Vendruscolo), use with care
s2re-fc                  = 100000
; Group to fit for S2 calculation
s2re-fitgrp              = backbone
```

6 Setting up and running a simulation

Use the script setup.sh to do the initial file conversion (from pdb to gromacs coordinate format) and the initial energy minimization of your molecule. Naturally, the molecule name and the force field should be adjusted in the script (OPLSAA is an all-atom force field, for NOE restraints an all-atom FF is needed because of the hydrogens!) This script requires a file named 'em.mdp' in the invocation directory (containing energy minimization parameters, see the original gromacs manual for more).

To run a simulation, use the script run.sh, adjust the molecule name and the number of replicas (parameter np) to your needs. This script requires the programs mdp_randomseed.pl (to give random initial velocities to each replica) and multigromacs.pl (a tool to ease up the invocation of the same command for multiple files). A run control file named full.mdp is also needed.

7 References

Best RB, Vendruscolo M (2004) Determination of Protein Structures Consistent with NMR Order Parameters. J Am Chem Soc 126:8090-8091

Richter B, Gsponer J, Várnai P, Salvatella X, Vendruscolo M (2007) TheMUMO (minimal under-restraining minimal over-restraining) method for the determination of native state ensembles of proteins. J Biomol NMR 37:117-135