Gromacs

get the source code  search the mailing list  online manual  git repository access

Sticky: Can't Post to the mailing list? Read the 'important information' section on the mailing list page.
Send a mail if you want to be registered as a contributor and be able to upload files or modify the webpage content.

News

September 2, 2010

The new release was a very short lived one - it was missing a few files that broke the CMake build system and now 4.5.1 is out.

September 1, 2010

It is finally here: Gromacs-4.5 has been just released. All critical issues have been resolved to deliver a stable and powerful package full of many new features.

Big thanks to all developers and users who made it happen!
NAMD, recipient of a 2002 Gordon Bell Award, is a parallel molecular dynamics code designed for high-performance simulation of large biomolecular systems. Based on Charm++ parallel objects, NAMD scales to hundreds of processors on high-end parallel platforms and tens of processors on commodity clusters using gigabit ethernet. NAMD uses the popular molecular graphics program VMD for simulation setup and trajectory analysis, but is also file-compatible with AMBER, CHARMM, and X-PLOR. NAMD is distributed free of charge with source code. You can build NAMD yourself or download binaries for a wide variety of platforms. Our tutorials show you how to use NAMD and VMD for biomolecular modeling.

The 2005 reference paper Scalable molecular dynamics with NAMD has over 1000 citations as of March 2010.

Search all NAMD resources: Search NAMD web site and tutorials

Spotlight: Nerve Signals (June 2010)

Nerve cells, through their electrical signals, control actions and intelligence of higher organisms. The signals result mainly from potassium and sodium ion channels in the cells: when the cells are stimulated electrically, they send an all (in case of sufficient stimulation) or nothing signal to other nerve cells or organs like
## GROMACS 4.0 Online Reference

### General Programs

<table>
<thead>
<tr>
<th>Getting Started</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow Chart</td>
<td>anadock</td>
</tr>
<tr>
<td></td>
<td>demux.pl</td>
</tr>
<tr>
<td>File Formats</td>
<td>do_dssp</td>
</tr>
<tr>
<td></td>
<td>editconf</td>
</tr>
<tr>
<td>mdp options</td>
<td>eneconv</td>
</tr>
<tr>
<td></td>
<td>g_anelm</td>
</tr>
<tr>
<td>FAQ</td>
<td></td>
</tr>
<tr>
<td></td>
<td>g_analyze</td>
</tr>
<tr>
<td></td>
<td>g_angle</td>
</tr>
<tr>
<td></td>
<td>g_bond</td>
</tr>
<tr>
<td></td>
<td>g_bundle</td>
</tr>
<tr>
<td></td>
<td>g_chx</td>
</tr>
<tr>
<td></td>
<td>g_cluster</td>
</tr>
<tr>
<td></td>
<td>g_clustsize</td>
</tr>
<tr>
<td></td>
<td>g_confrms</td>
</tr>
<tr>
<td></td>
<td>g_covar</td>
</tr>
<tr>
<td></td>
<td>g_current</td>
</tr>
</tbody>
</table>
Programs by Topic

Generating topologies and coordinates
Running a simulation
Viewing trajectories
Processing energies
Converting files
Tools
Distances between structures
Distances in structures over time
Mass distribution properties over time
Analyzing bonded interactions
Structural properties
Kinetic properties
Electrostatic properties
Protein specific analysis
Interfaces
Covariance analysis
Normal modes

Generating topologies and coordinates
pdb2gmx converts pdb files to topology and coordinate files
x2top generates a primitive topology from coordinates
editconf edits the box and writes subgroups
genbox solvates a system
genion generates mono atomic ions on energetically favorable positions
genconf multiplies a conformation in 'random' orientations
Generating topologies and coordinates

- **pdb2gmx**: converts pdb files to topology and coordinate files
- **x2top**: generates a primitive topology from coordinates
- **editconf**: edits the box and writes subgroups
- **genbox**: solvates a system
- **genion**: generates mono atomic ions on energetically favorable positions
- **genconf**: multiplies a conformation in 'random' orientations
- **genrestr**: generates position restraints or distance restraints for index groups
- **protonate**: protonates structures

Running a simulation

- **grompp**: makes a run input file
- **tpbconv**: makes a run input file for restarting a crashed run
- **mdrun**: performs a simulation

Viewing trajectories

- **ngmx**: displays a trajectory
- **trjconv**: converts trajectories to e.g. pdb which can be viewed with e.g. rasmol

Processing energies

- **g_energy**: writes energies to xvg files and displays averages
- **g_enemattr**: extracts an energy matrix from an energy file
- **mdrun**: with -rerun (re)calculates energies for trajectory frames
genbox

Main Table of Contents

Description

Genbox can do one of 3 things:

1. **Generate a box of solvent**. Specify `-cs` and `-box`. Or specify `-cs` and `-cp` with a structure file with a box, but without atoms.

2. **Solvate a solute configuration**. eg., a protein, in a bath of solvent molecules. Specify `-cp` (solvent) and `-cs` (solute). The box specified in the solute coordinate file (`-cp`) is read, unless `-box` is set. If you want the solute to be centered in the box, the program `editconf` has sophisticated options to change the box dimensions and center the solute. Solvent molecules are removed from the box where the distance between any atom of the solute molecule(s) and any atom of the solvent molecule is less than the sum of the Vander-Waals radii of both atoms. A database of Vander-Waals radii is read by the program, atoms not in the database are assigned a default distance `vdw`.

3. **Insert a number (-nml) of extra molecules (-co) at random positions**. The program iterates until `nml` molecules have been inserted in the box. To test whether an insertion is successful the same Van-Der-Waals criterion is used as for removal of solvent molecules. When no appropriately sized holes (holes that can hold an extra molecule) are available the program tries for `-nml` * `-try` times before giving up. Increase `-try` if you have several small holes to fill.

The default solvent is Simple Point Charge water (SPC), with coordinates from `genlib/spc216.gro`. Other solvents are also supported, as well as mixed solvents. The only restriction to solvent types is that a solvent molecule consists of exactly one residue. The residue information in the coordinate file is used, and should therefore be more or less consistent. In practice this means that two solvent molecules in the solvent coordinate file should have different residue number. The box of solute is built by stacking the coordinates read from the coordinate file. This means that these coordinates should be equilibrated in periodic boundary conditions to ensure a good alignment of molecules on the stacking interfaces.

The program can optionally rotate the solute molecule to align the longest molecule axis along a box edge. This way the amount of solvent molecules necessary is reduced. It should be kept in mind that this only works for short simulations, as eg., an alpha-helical peptide in solution can rotate over 50 degrees, within 500 ps. In general it is therefore better to make a more or less cubic box.

Setting `-shell` larger than zero will place a layer of water of the specified thickness (nm) around the solute. Hint: it is a good idea to put the protein in the center of a box first (using `editconf`).

Finally, genbox will optionally remove lines from your topology file in which a number of solvent molecules is already added, and adds a line with the total number of solvent molecules in your coordinate file.