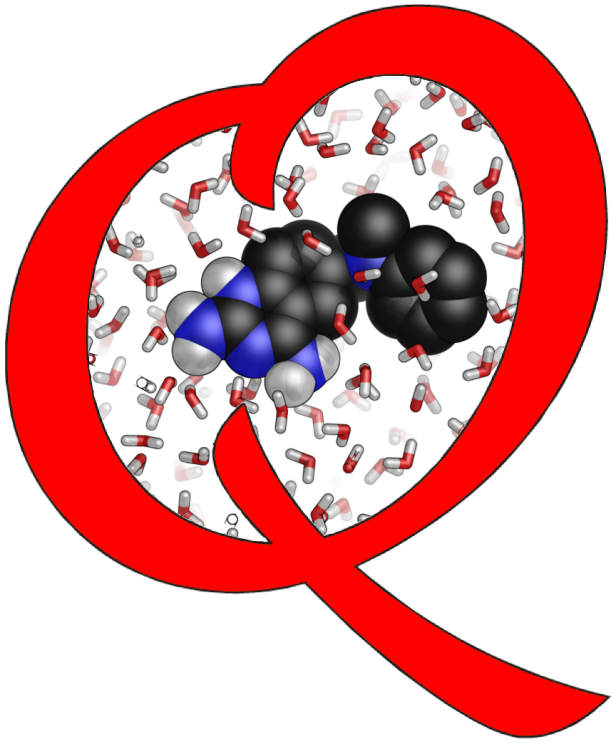

Manual for the Molecular Dynamics package Q

version 5.06
January 1, 2004
March 26, 2014
Q web site: http://xray.bmc.uu.se/aqwww/Q

Contents

A INTRODUCTION

Molecular dynamics (MD) simulation can be used to sample the thermally accessible region of conformational space of a microscopic model of a molecular system. From the ensemble of sampled structures and their associated potential energies (given by the force field or molecular mechanics potential energy function) it is, in principle, possible to calculate free energies. Quantities such as binding free energies, solvation free energies and activation free energies are particularly interesting to calculate because they are the direct result of thermodynamic or kinetic experiments. It is thus possible both to quantitatively verify calculated results against experimental data and to make predictions which can be tested experimentally.

Q [1] is a set of tools for such calculations, tailored for some specific kinds of free energy calculations, namely, (I) free energy perturbation (FEP) simulations [2, 3], (II) empirical valence bond (EVB) calculations [4, 5] of reaction free energies and (III) linear interaction energy (LIE) calculations[6, 7, 8] of receptor-ligand binding affinities.

The main features which distinguish Q from other MD packages are:

- **The spherical boundary.** Q is intended for free energy calculations in biomolecular systems solvated in a spherical droplet of explicit water molecules. Using a spherical boundary [9, 10, 11] makes it possible to limit the size of the simulated system, *i.e.* to focus the simulation on a smaller region such as a binding site, and also makes accurate treatment of long-range electrostatics rather inexpensive.
- **The flexibility in choice of force field.** The force fields are defined in parameter files, separate from the program and the choice of force field is thus simply a matter of which parameter file to use.
- **The ease of use and learning.** The simulation control input and force field definition files are organised in a flexible way and easy to understand and modify. The programs give extensive diagnostics when problems are encountered.
- **It runs on any computer and simulates any number of particles.** By using dynamic memory allocation Q can simulate biomolecules of moderate size on a personal computer, or very large molecular systems on a super-computer, without any modifications of the program.

Two new features that have been added in version 5.0:

- **Periodic boundary condition.** The periodic boundary allows an additional way to perform calculations.
- **Parallel version.** By using more than one computer a single simulation can be executed faster. The parallel version is especially useful when simulating large systems, *e.g.* the systems that are used with the periodic boundary condition.

B USER GUIDE

The structure of Q resembles that of many other MD simulation programs. The main program is **qdyn** which carries out the actual trajectory calculations. Besides normal input control data, it needs the type of data usually referred to as the molecular topology. This is a file containing information about how atoms are bonded to each other etc. together with all the parameters of the force field (FF) to be used. This topology file is created with the preparation program Qprep which is an interactive program that uses pdb (Protein Data Bank) coordinate files together with FF specific files to generate the topology. Qprep can also be used for various other data transformation purposes as described below. Input data for FEP, EVB or receptor-ligand complex simulations is given in a separate file, referred to below as the fep file. The fep file lists atoms to be transformed, called Q-atoms and force field parameters for the different states in perturbation simulations. For analysis of the computations the main tool is Qfep which is a program that carries out FEP and EVB calculations of free energies utilising energy data produced by **qdyn**. A number of utility programs (trajectory and energy averaging, radial distribution functions, ...) are also provided. The general outline of Q is shown in figure 1.

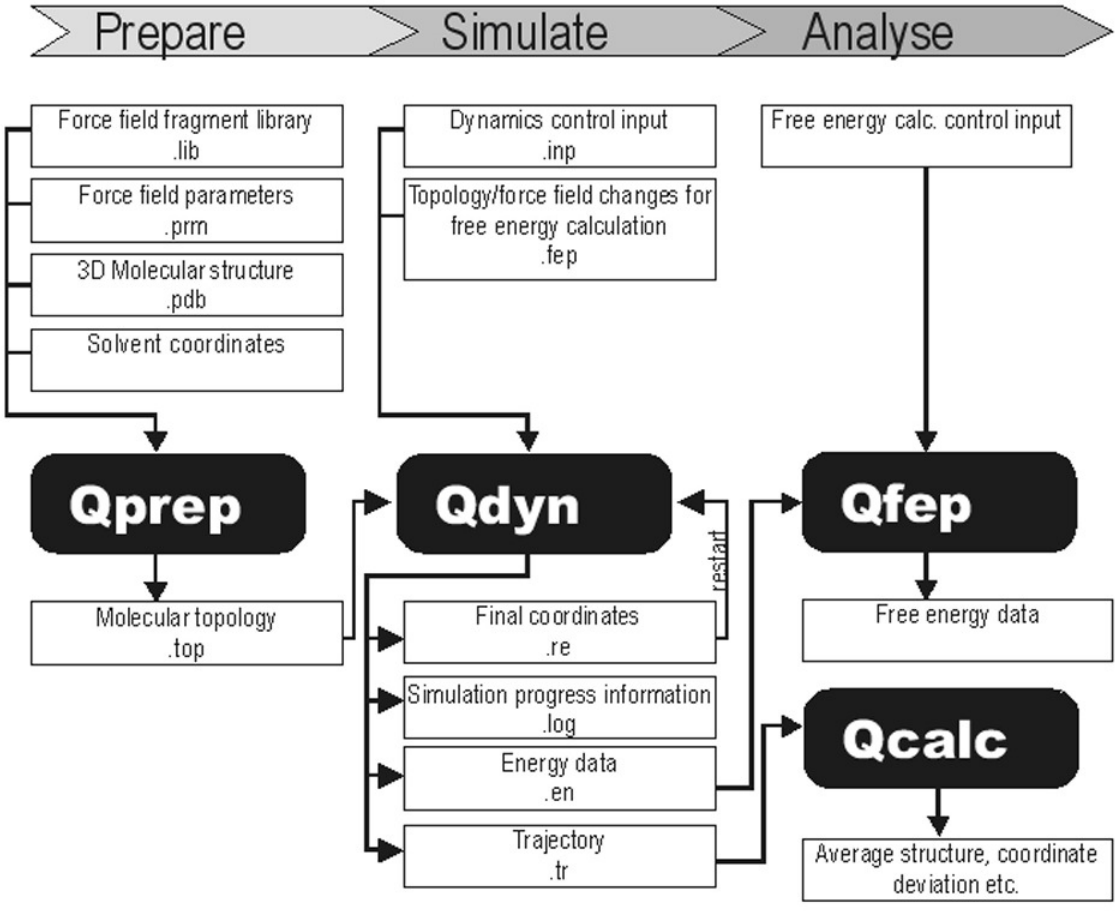

Figure 1: Overview of the procedure for free energy calculation with Q. The white boxes represent files and also show typical file name extensions. The black boxes are programs.

In the following sections we will go through the normal sequence of steps for preparing a topology file with Qprep and then describe in detail the input data required for running MD simulations with **qdyn**.

B.1 Preparing a molecular topology

The topology file prepared with the program Qprep contains all the information about the molecular system needed for a simulation with **qdyn**. To make a topology you need

- A molecular fragment library file describing the atoms and connectivity of each protein residue, ligand etc., collectively referred to as library entries.
- A force-field parameter file.
- Coordinates for all (non-solvent) atoms in the form of a PDB file.

B.1.1 Preparing coordinates

The PDB file format [12] used for co-ordinate input to Qprep is a fixed format, *i.e.*, the number of spaces between columns of data is significant and tab characters are not permitted. Case is significant - lower-case letters should not be used. Only ATOM and HETATM records are read by Qprep, all other records are ignored.

Hydrogen atoms are not required - their coordinates will be generated by Qprep in such a way that the bond and angles to the hydrogen atom are at an energy minimum. To further control the placement of hydrogens, a special torsion potential defined in the fragment library entry may be included. See [**build_rules**] on page 46. If hydrogen atom coordinates are given in the PDB file, they will not be altered.

When a structure containing more than one molecule is loaded into Qprep, the program will identify the boundaries between molecules either by the type of fragment or using 'gap' marker lines. Fragments which are monomers in a polymer chain, like amino acids, have designated 'head' and 'tail' atoms in their library entries, defining how they should be connected to the neighbouring residues. Library entries describing separate molecules such as solvent or ligands do not contain this linkage information. To distinguish two peptide chains from each other it is necessary to introduce a gap marker line in the PDB file after the last atom of the first molecule. The gap marker consists of the word GAP in capital letters on a separate line.

The numbering of atoms in the PDB is not significant. Note that Qprep will renumber all atoms in a single sequence starting at one. This is necessary in order to incorporate hydrogen atoms in the sequence. The result is thus that *atom numbers in the PDB file read by Qprep and PDB files generated by Qprep will be different.* Residue numbers are used merely to distinguish one residue from the next and the residues will also be renumbered by Qprep. Residue numbers must be numeric, alphanumeric identifiers such as 60A sometimes encountered in the protein data bank are not permitted.

Protein structures determined by X-ray crystallography are normally refined with a large number of water molecules. Some of these well-ordered water molecules may be important

for the structure and function of the protein and should be included in the simulation. However, most of the crystallographic waters surrounding the protein can be removed. In fact, the presence of a large number of water molecules around the protein surface will disturb the solvation algorithm leading to inhomogenous water density. Qprep assumes that solvent molecules appear after solute molecules in the PDB file and keeps track the last solute atom. Solvent molecules are identified by their residue names. The list of solvent residue names is a user-setable preference in Qprep (see Setting preference on page 9) with the default value WAT, HOH, H2O, SPC, TIP3. Solvent molecules added by the solvation algorithm will appear at the end of the atom sequence. Solvent molecules outside the simulation sphere (more than water radius + 2 Å from solvent centre) will be excluded like solute atoms outside the solute sphere.

B.1.2 Selecting a force field

Q is designed to run with a wide selection of force fields. Fragment libraries and force field parameters for AMBER95, AMBER/OPLS, OPLS-AA, CHARMM v.22, GROMOS87 and GROMOS96 are supplied in with the program. The corresponding library and FF parameter files are listed in table 7 on page 41. In addition to all the interaction parameters, a Q force field parameter file also include overall properties of the force field, such as the van der Waals parameter combination rule. Selecting a force field is thus equivalent to loading the appropriate fragment library and parameter file into Qprep.

B.1.3 Adding force field library entries

Your simulations will probably include molecular fragments other than the amino acid residues and solvent molecules that are included in the standard library files and you therefore need to write a library entry for each 'new' residue, ligand, co-factor or other molecule. We suggest you keep your library entries in a separate file rather than adding to the standard library file. Load first the standard library and then your own into Qprep. If you need to modify an entry in the standard library, add the modified entry (keeping the same name) to your own library file - it will replace the old entry when loaded.

The best starting point for generating a new entry is to copy an old one, this ensures the correct syntax. For details, see the section Fragment library file format on page 45.

B.1.4 Running Qprep

Qprep is a command-oriented program designed to be run interactively. The first thing you need to learn about Qprep is that there is a help command which gives a list of available commands. The program's input parser will prompt for the parameters required for each command but will also accept complex command lines including the parameters. Thus, you may either enter the command **readlib** and await the prompt 'Name of molecular library' or you may type **readlib my_ligands.lib** on one line.

A good deal of effort has been spent to make Qprep handle errors gracefully. When a problem is encountered in a fragment library, parameter file or PDB file, you will be

notified but the processing of the file will proceed, thus exposing also errors later in the file. After correcting the problem you simply read the file again without the need to restart the program.

The different steps to generate a topology with Qprep are described below, in the sequence they are normally executed. More detailed information is available in the section Topology preparation reference on page 42.

Loading fragment libraries

Use the **readlib** command to load fragment libraries. Many libraries may be loaded by repeating the **readlib** command for each of them. Note that if one entry name occurs more than once, the latest definition takes precedence. A warning message is issued when a library entry is overloaded.

Errors encountered while reading a library will be displayed and after they have been corrected, the library may be loaded again. To remove all library entries from memory, use the **clearlib** command.

Loading force field parameters

Use the **readprm** command to load force field parameters. As opposed to the case with fragment libraries, only a single parameter file can be loaded. If you need to modify the parameter file, edit and save it and load it again. It is not necessary to reload fragment libraries or the structure.

Loading coordinates

Use the **readpdb** command to load the molecular structure file. Before loading the structure, the program will verify that all the required library entries are loaded and that the number of heavy atoms of each fragment is correct. Only one file can be loaded - reading another file clears the previously loaded structure. This is useful as you reload the same file after correcting a problem.

Choosing boundary condition

Use the **boundary** command to set the boundary condition; sphere or box. When a boundary has been chosen the centre and radius in the spherical case, and boxlengths in the periodical case are specified. It is important that the boundary condition defined in the topology file is consistent with the boundary condition used when running dynamics with **qdyn**.

Adding solvent

Use the **solvate** command to add solvent molecules to the loaded structure. Using spherical boundary solvent can be generated by any of the following methods:

- Using randomly oriented molecules on a grid. This option does not depend on any solvent coordinate file but only asks which library entry should be used. The density is specified in the library entry.
- By reading solvent coordinates from a solvent file. The solvent file is similar to a PDB file, see Solvent file format on page 51. The solvent file contains coordinates for

a sphere or box of solvent. In the case of a box, it will be replicated in all directions as needed and can thus be used to solvate a system of any size.

- By reading solvent coordinates from a restart file from a previous simulation of the same molecular system.

When using one of the two first methods, Qprep first fills a sphere completely with solvent and then deletes molecules where heavy atoms are closer than a threshold distance to any heavy atom in the loaded structure. The threshold distance is controlled by the Qprep preference value **solvent_pack**. Crystallographic solvent molecules (in the PDB file loaded by readpdb) more than 2 Å outside the solvent sphere will be excluded from the simulation to avoid excessive radial restraining forces.

If periodic boundary is used it is not possible to use a solvent file with a sphere of solvent. Moreover the threshold distance is also applied to solvent molecules near the boundary, to avoid crashes between solvent molecules in neighbouring boxes.

At present, the choice of solvent is limited to tri-atomic molecules like the SPC[13] and TIP3P[14] water models or methanol in an extended atom representation.

Note: Make sure that crystallographic water molecules and waters added by the solvate command have *the same residue name*! Otherwise Qprep will mark the topology as mixed-solvent, which is not implemented in **qdyn** at present.

Adding cross-link bonds

Cross-link bonds such as disulphide bridges in proteins are not generated automatically they are not defined in the fragment library. Extra bonds can be added automatically by searching the solute molecules for close but not bonded atom pairs, or manually by specifying atom numbers. To generate cross-link bonds automatically, use the **xlink** command. For each close atom pair found, you need to confirm or reject the making of a bond. To add bonds manually, use the **addbond** and specify the atoms to be bonded, either by atom numbers or in the form residue_number:atom_name.

Generating the topology

The **maketop** command is used to create the topology in memory. The only input for this command is a name for the topology. The process involves the following steps which are all carried out by the maketop routine without further user input:

- Setting atom types and partial charges.
- Generating the lists of bonds, bond angles, torsion angles and improper torsions.
- Generating the neighbour exclusion and 1-4 neighbour lists.
- Generating coordinates for hydrogen atoms not included in the loaded structure.
- Marking atoms outside the simulation sphere as excluded
- Calculation of the 'effective solvent radius' used in the simulation to ensure correct density of the solvent. This radius which is based on the number and spatial distribution of solute and solvent atoms in the sphere, typically differs somewhat from the radius used in the solvation step.

Verifying the topology

The successful generation of a topology in the step above is no guarantee that it is correct. Fortunately Qprep offers a number of commands to check the topology:

- **Checkbonds** is used to list bonds with a potential energy exceeding a specified threshold. This helps to identify errors in the connectivity.
- **Checkangs** lists angles with energies over a threshold.
- **Checktors** lists torsion angles with energies over a threshold.
- **Checkimps** lists high energy improper torsion angles. This is a very important step for users of force fields with harmonic improper potentials (GROMOS, charmm) since with a non-periodic potential it makes a huge difference if the angle is *e.g.* -179^o instead of +179^o if the (single) energy minimum is at +180^o although the structural difference is small. *Impropers with the wrong sign give rise to high energies and strong forces which will distort the molecule during the simulation and must be corrected.* (This is not a problem with periodic improper torsion potentials used in the other force fields.) The sign of the angle depends on the order of the bonds in the fragment library entry, each permutation of the sequence of the bonds involved will change the sign of the angle. Instead of modifying library entries, reloading the libraries and remaking the topology, the **changeimp** command can be used to flip the sign of selected impropers or of all impropers with energy exceeding a threshold value.

Writing topology and coordinate files

The final step in making a topology is saving it to a file by using the **writetop** command.

You will also need to make a PDB (or mol2, see below) file from the topology to see the new numbers of the atoms. These are the atom numbers you need to refer to when setting up restraints and topology modifications for perturbation simulations. Use the **writepdb** command to write a PDB file containing all the atoms of the topology.

Setting preferences

A number of parameters that affect the operation of Qprep, *e.g.* during solvation, but which are not normally changed are not required as input to the commands. These parameters may be changed by experience users by the preference mechanism in Qprep.

The **prefs** command is used to list the values of user-setable parameters and the **set** command to change a value.

The preference parameters and their default values are listed in the table on page 45.

B.2 Running dynamics with qdyn

Once a molecular topology file has been generated with Qprep, you can carry out MD, FEP and EVB simulations with **qdyn**. The simulation can have either spherical or periodic boundary condition, and can be executed either sequentially or in parallel.

In this section we will describe the basic functions of **qdyn** and go through the different options that are available for control of the dynamics runs. There are two main input files

that are used to set up the dynamics specifications:

1. The **qdyn** input file that controls things like time-step, temperature, cut-offs, restraints etc.
2. The FEP file which is an auxiliary file whose function is to redefine the topology information for certain atoms. This enables the explicit control over selected force field parameters that is necessary for FEP and EVB calculations.

B.2.1 Simulation procedure

The MD simulation required for a free energy calculation often proceeds in multiple stages. Normally, the initial stage is run at a very low temperature with strong coupling to the temperature bath (similar to energy minimisation) to relax strain in the initial structure. Then may follow stepwise heating of the simulated system and equilibration for some time at the target temperature. After this comes the main simulation during which energy and structure data is collected. For perturbation simulations, this phase is composed of a series of simulations using intermediate potentials defined by different sets of weight coefficients for the FEP states.

A separate **qdyn** input file is used for each sub-simulation. It is therefore practical to prepare a command file (shell script or batch file) which executes all the sub-simulations sequentially. The name of the input file is passed to **qdyn** as the first (and only) argument on the command line. (It is not possible to use redirection of the standard input stream by the < operator.) A simple example of such a file where **qdyn** is invoked once for each input file and the output redirected to a log file follows:

Table 1: Multi-stage simulation command file

qdyn	relax.inp	>	relax.log
qdyn	eq1.inp	>	eq1.log
qdyn	eq2.inp	>	eq2.log
qdyn	eq3.inp	>	eq3.log
qdyn	data1.inp	>	data1.log
qdyn	data2.inp	>	data2.log

B.2.2 Output generated by qdyn

The different data files generated by **qdyn** are (shown in the overview in Figure 1):

- General information about the progress of the simulation including energy summaries and temperature is written to the standard output device and normally redirected to a log file.
- Final coordinates and velocities are written to a 'restart' file to be used to start the next sub-simulation and, after conversion to a structure file (see Analysing structures from the simulation on page 26, for viewing the final structure. This file is also updated during the simulation and if the forces and velocities become too large and

the simulation is terminated prematurely. The file thus also serves a diagnostic purpose.

- Energy data for Q-atoms in each FEP state is written to an energy file every few time steps (determined in the input file).
- Coordinates for all or a subset of atoms are written to a trajectory file at regular intervals (determined in the input file).

The restart and energy files are Fortran binary files. The trajectory file follows the DCD format also used in other MD programs (Charmm, X-plor) and can be read by many visualisation and trajectory animation programs.

B.2.3 Preparing qdyn input files

In this overview the various aspects of defining the simulation set-up are introduced. For more complete information, see **qdyn** input file format on page 54.

The **qdyn** input file contains the specification of the dynamics simulation. It is a text file divided into sections, starting with a section heading and containing information on the different aspects of the simulation. The sections are of two kinds:

- Sections where each line consists of a keyword and a values, with different keywords on each line.
- Sections where all the lines have the same formatting and together constitute a data set. No keywords are used here.

Only a few sections are mandatory, most are optional and they may appear in any order. The formatting within a section is flexible in that blank lines are permitted as well as comments (starting with !, # or *) at the end of lines or on separate lines. The format of the data in each record within a section is free (white space is not significant), but all data in the record must be on the same line.The units used are based on Å, K and kcal/mol (see Units on page 54).

Dynamics control information

The section [**MD**] is normally the first in the input file the most apparently required section, since it defines the core parameters of the simulation like the number of time steps, their size and the temperature. Below is an example of a basic MD section in an input file:

```
[MD]
steps          10000
stepsize       2.0
temperature    300
shake_solvent  on
shake_solute   off
lrf            on
```

Periodic boundary conditions

The [**PBC**] section contains options and settings for simulations with periodic bound-

ary. The mere existence of the section header [**PBC**] is enough to indicate the boundary condition. Additional options is added as exemplified below:

```
[PBC]
rigid_box_centre      on
constant_pressure     on
max_volume_displ      65
pressure              1.5
```

The rigid_box_centre option gives a periodic box with fixed coordinates instead of centering the box around the solute. In the above example the Monte-Carlo constant pressure algorithm, described in [17], is performed with target pressure 1.5 bar and maximum volume displacement 65 $Å^3$.

Non-bonded interactions

Cut-off radii for the non-bonded interactions for different categories of atoms are given in the section [**cut-offs**], as exemplified below:

```
[cut-offs]
solute_solute     10
solvent_solvent   10
solute_solvent    10
q_atom            10
```

The q_atom entry defines the cut-off for interactions between Q-atoms and non-Q-atoms. No cut-off is used for interactions among Q-atoms. When using periodic boundary conditions, make certain all cut-off radii are less than half the shortest boxlength.

Sphere

The **sphere** section defines parameters concerning the spherical boundary. The most frequently used parameter is the shell_radius that allow the user to restrain solute atom in a shell to their original coordinates as defined in the topology.

```
[sphere]
shell_radius   18   !Restrain solvent in inner shell
shell_force    10   !Restraining force constant
```

Solvent

The [**solvent**] section controls the solvent boundary restraints when simulating with a spherical boundary. This section is thus omitted when periodic boundary is used. It is possible to fine-tune the restrains, but the default values used if no data is given are adequate for most simulations. The contents may often be as simple as:

```
[solvent]
polarisation   on   !Enable solvent polarisation restraints
```

Update and data collection intervals

The frequencies of regular events in the simulation are defined in the section [**intervals**]. These events are the regeneration of the non-bonded pair lists and the writing of energies or coordinates to the energy, trajectory and output files. Example:

```
[intervals]
non_bond    25
output      5
energy      0     !No energy file
trajectory  100
```

This example specifies that the non-bonded pair lists should be regenerated every 25 time steps, energy summaries written to the terminal or log file every 5 steps, no energy file is to be written and coordinates written to the trajectory every 100 steps.

Trajectory

If the coordinates of only a subset of the atoms are to be stored in a trajectory file, the selection of atoms is done in the section [**trajectory_atoms**], which could look as follows:

```
[trajectory_atoms]
heavy not excluded residue  1    104
residue                     105  106
residue                     109
```

In this atom mask the heavy atoms of residues 1 to 104 which are inside the simulation sphere and all atoms of residues 105-106 and 109 are selected. For further information see Atom masks on page 42.

Files

The names of files to be read and written are grouped together in this section. A topology file and a name for the final coordinates file must always be specified here. A restart file may be specified to start the simulation using the final coordinates and velocities from a previous simulation of the same system. For perturbation simulations the name of an FEP file is needed. If trajectory and energy files should be generated they need to be named here.

```
[files]
topology    molecule.top
final       data_01.re
trajectory  data_01.dcd
energy      data_01.en
fep         molecule.fep
```

FEP state weight coefficients λ

For multi-state perturbation simulations the mapping vector λ whose components are the weight coefficients for the FEP states is given on a single line under the section heading [**lambdas**]. For a simple two-state mapping potential with 70% of state 1 and 30% of state 2 it would look like this:

```
[lambdas]
0.70 0.30
```

Restraints

Several types of geometrical restraints can be applied to the simulated system to eliminate large movements, maintain interatomic distances or to stop the diffusion of a solute towards

the sphere boundary. The most straight-forward type of restraints are harmonic potentials applied to restrain a sequence of atoms to their initial coordinates (in the topology file). This type of restraining is specified in the [**sequence_restraints**] section and requires only the number of the first and last atom of the sequence and a force constant. The restraints may be applied to heavy atoms only or to all atoms in the sequence. Instead of restraining each atom individually to its initial position, the set of atoms can be restrained as a whole to its initial geometrical centre. In this case identical forces are applied to all the atoms. This alternative, used with a low force constant, is useful *e.g.* to keep a small solute molecule at the centre of the simulation sphere without hindering its tumbling motion (rotation). Both variants are exemplified below:

```
[sequence_restraints]
21   40   5.0   0
65   72   2.0   1   1
```

Here, atoms 21 to 40 are restrained to their initial positions by 5.0 kcal·mol^{-1}·Å^{-2} but hydrogens are excepted (0). Atoms 65 to 72 including hydrogens (1) are restrained as a group to their initial geometrical centre (1).

Restraints on individual atoms are not restricted to use the initial position as a reference since the "target" position is specified in the input. The restraint may be applied only in a single FEP state or in all states. In the first case the force is scaled by the weight coefficient λ for that state. Different force constants may also be used for the x, y and z axes. By setting one or two force constants to zero, the atom will be restrained to a line or a plane, respectively. An example of an [**atom_restraints**] specification follows:

```
[atom_restraints]
8   82.5   28.32   72.6   5.   5.   5.   0
```

In this case atom 8 is restrained to the point (x, y, z) = (82.5, 28.32, 72.6) with 5.0 kcal·mol^{-1}·Å^{-2} along all axes in all FEP states (0).

The distance between two atoms may be restrained using either a standard harmonic potential or a flat-bottomed harmonic well potential, by adding an entry under the heading [**distance_restraints**] as follows:

```
[distance_restraints]
13   20   4.5   5.0   10.0   1
```

Atoms 13 and 20 are here held together by a flat-bottomed harmonic well potential which is zero between 4.5 and 5.0 Å and has a force constant of 10.0 kcal·mol^{-1}·Å^{-2} for other distances. It is active in FEP state 1 only.

Another means of restricting the overall motion of a molecule (when using spherical boundary) is to apply a soft-wall or half-harmonic restraint outside a given radius from the (solvent) sphere centre. This is done in the section [**wall_restraints**] *e.g.*:

```
[wall_restraints]
80    99    14.0   5.0   0   0   0
102   102   14.0   5.0   0   0   0
```

In this example atoms 80 to 99 and 102 will experience an inward harmonic force if they

are beyond 14 Å from the sphere centre. The force constant is 5.0 kcal·mol^{-1}·Å^{-2} but force will not be applied to hydrogen atoms (last 0). D_e is the depth of the Morse potential and a is the exponential coefficient of the Morse term. For obvious reasons the [**wall_restraints**] section is not used in combination with periodic boundary conditions. One can choose between harmonic or Morse potential.

B.2.4 qdyn input file examples

We give two annotated examples below. The first is the simplest possible input file, using default values for all optional parameters. The second is a bit more elaborate and exemplifies the use of many extra options such as special restraints. Detailed information about the data in each section is found in the section **qdyn** input file format on page 54.

Table 2: Minimal **qdyn** input file

Data		**Description**
[MD]		Basic data for the simulation
steps	2000	Number of steps
stepsize	1.0	Step size (fs)
temperature	1	Temperature (K)
initial_temperature	1	Temperature (K) for Maxwell-distributed initial velocities
[files]		File names for input and output
topology	molecule.top	Topology file
final	molecule.re	Restart to write at end of simulation

Table 3: Advanced **qdyn** input file

Data		**Description**
[MD]		Basic data for the simulation
steps	10000	Number of steps
stepsize	2.0	Step size (fs)
temperature	300	Temperature (K)
bath_coupling	10	Temperature bath relaxation time (fs)
random_seed	57643	Seed for random number generator (only for initial vel.)
initial_temperature	300	Temperature (K) for Maxwell-distributed initial velocities
shake_solvent	on	Shake bonds & angles of water
shake_hydrogens	on	Shake bonds to hydrogen in solute & solvent
lrf	on	Use lrf for electrostatics beyond cut-off
[cut-offs]		Cut-off radii for different groups of atoms
solute_solute	10	Solute-solute cut-off (Å)
solvent_solvent	10	Water-water cut-off (Å)
solute_solvent	10	Solute-water cut-off (Å)

Table 3: Advanced **qdyn** input file

Data		**Description**
q_atom	10	Q-atom non-q-atom cut-off (Å)
[sphere]		Definition of the simulation sphere
shell_radius	18	Definition of the inner restrained shell (Å).
shell_force	10.0	Restraining force constant in shell (kcal·mol^{-1}·Å^{-2})
[solvent]		Solvent boundary settings
radial_force	60.0	Force constant for radial restraining (kcal·mol^{-1}·Å^{-2})
polarisation	on	Use polarisation restraints (this is the default)
polarisation_force	20.0	Force constant for polarisation restraining (kcal·mol^{-1}·rad^{-2})
[intervals]		Intervals for saving data
non_bond	25	Interval for generation of non-bond lists (steps)
output	5	Interval for energy summary in output
energy	10	Interval for energies to energy file
trajectory	100	Interval for coordinates to trajectory file
[trajectory_atoms]		Select atoms to be included in the trajectory file
heavy not excluded residue	1 104	Select heavy atoms of residues 1 to 104 which are not excluded
residue	105 106	Select all atoms of residue 105 to 106
residue	109	Select all atoms of residue 109
[files]		File names for input and output
topology	molecule.top	Topology file
final	data_01.re	Restart to write at end of simulation
trajectory	data_01.dcd	Trajectory file to write
energy	data_01.en	Energy file to write to
fep	molecule.fep	FEP file
[lambdas]		Weights for the FEP states
0.70 0.30		Lambda value for each state
[sequence_restraints]		Restrain contiguous sequences of atoms to initial coordinates
21 40 5.0 0		First & last atom, force const. (kcal·mol^{-1}·Å^{-2}), H-flag
65 72 2.0 1 1		First & last atom, force const., H-flag, restrain-to-centre-flag
[atom_restraints]		Individual atom positional restraints
8 2.5 8.3 7.6 5. 5. 5. 0		atom, x0,y0,z0, fcx, fcy, fcz, FEP state (0=all)
[distance_restraints]		Atom-atom distance restraints
13 20 4.5 5.0 10.0 1		Atom i, atom j, lower r, upper r, fc, FEP state (0=all)
[wall_restraints]		Half-harmonic (elastic wall) sequence restraints
80 99 14.0 5.0 0 0 0		First & last atom, r0 (from water centre), fc, D_e (kcal· mol^{-1}), a (Å^{-1}), H-flag

Table 3: Advanced **qdyn** input file

Data	**Description**
102 102 14.0 5.0 0 0 0	First & last atom, r0 (from water centre), fc, D_e, a, H-flag

B.2.5 FEP file

The purpose of the FEP file is to define a set of atoms as Q-atoms and to redefine their interaction parameters. All kinds of force-field parameters for these atoms can be controlled and several different "states" can be defined. The parameters for the different states may differ very little, *e.g.*, in the van der Waals parameters of a single atom, or the states can represent different valence bond structures. A typical application of the latter case would be to model reactants and products of a chemical reaction to be investigated by EVB simulation as two different states or, for a multi-step reaction, one state for the products of each elementary reaction step. In such a model of a reaction bonds, angles, torsions, partial charges, vdW parameters etc. may change for many atoms.

The idea behind this definition of different states is that **qdyn**, for each configuration of the system's particles, will keep track of the energies of each state and write these to the energy file. The mapping potential or sampling potential used to generate the forces controlling the dynamics is a mixture of the FEP/EVB states, determined by the mapping parameter or weight coefficient λ given to each (pure) state in the **qdyn** input file. The free energy differences between FEP/EVB states can then easily be calculated by Qfep using the standard FEP formula or the potential of mean force (umbrella sampling) approach to obtain the EVB ground state reaction free energy profiles.

The FEP file has the same overall structure as the **qdyn** input file (see page 11) with various kinds of data grouped into sections, the majority of which are optional. We will describe FEP files for a couple of prototype cases, starting with the simpler ones. For a complete description of the file format, see FEP file format on page 58.

Example: Charging a benzene molecule

The FEP file shown below may be used to calculate the electrostatic contribution to the free energy of solvation for a benzene molecule. The atoms and bonds of the molecule are defined in a topology file (not shown). In our topology the carbon atoms have odd numbers and hydrogens have even numbers.

Data	**Description**
[FEP]	Free energy perturbation
states 2	No. of states
[atoms]	Designate atoms in topology as q-atoms
1 1	
2 2	
3 3	
4 4	
5 5	

Data			**Description**
6	6		
7	7		
8	8		
9	9		
10	10		
11	11		
12	12		
[change_charges]			Assign new charges for each state
1	- 0.15	0.0	Q-atom no., charges in state 1 & 2
2	+0.15	0.0	
3	- 0.15	0.0	
4	+0.15	0.0	
5	- 0.15	0.0	
6	+0.15	0.0	
7	- 0.15	0.0	
8	+0.15	0.0	
9	- 0.15	0.0	
10	+0.15	0.0	
11	- 0.15	0.0	
12	+0.15	0.0	

The value following the keyword states in the section [**FEP**] is the number of FEP/EVB states. In the [**atoms**] sections atoms from the topology are designated as q-atoms. The first column of the data records in this section is the q-atom number given to the atom (used later to refer to it) and the second column is the number of the atom in the topology. The data in the section [**change_charges**] defines the charge of q-atoms in each state. Here we are changing the charges of all atoms, but in general only the charges which change need to be listed.

In the case above we have made no changes to the bonded or vdW parameters of the benzene molecule and the FEP file is simply used to define two "charge" states, one with the CH dipolar charges being ±0.15 e and one state with zero partial charges.

Example: Changing van der Waals parameters

In this example we will take a look at how to redefine van der Waals (Lennard-Jones) interaction parameters. The FEP file shown below may be used if we want to calculate the difference in hydration free energy between two ions, in this case Na^+ and K^+. Since the ions have the same charge the only change that needs to be made in a perturbation calculation between the two ions is to define two sets of Lennard-Jones interaction parameters.

Table 4: FEP file for perturbation of Na^+ to K^+.

Data		**Description**
[FEP]		
states	2	No. of states
[atoms]		Designate atoms in topology as q-atoms

Table 4: FEP file for perturbation of Na^+ to K^+.

Data					**Description**		
1	1				Q-atom no., topology atom no.		
[atom_types]					Define new atom types (LJ parameters,)		
!Name	Ai	Bi	Ci	ai	Ai(1-4)	Bi(1-4)	Mass
Na	143.70	3.89	0.0	0.0	0.0	0.0	22.99
K	522.70	4.35	0.0	0.0	0.0	0.0	39.10
[change_atoms]					Assign new atom types to Q-atoms		
1	Na	K			Q-atom no., q-atom type in states 1 and 2		

Here we again define two states, but now only for one Q-atom that has number 1 in our simple topology file which only contains the single ion. No charges need to be changed since both ions are monovalent cations, and the section [**change_charges**] is therefore omitted. The only specific definitions needed here are the following. In the section [**atom_types**] the parameters for the atoms involved in the perturbation are given. Whether (A_i, B_i) or (R^*, ε) LJ parameters are used depends on the combination rule specified in the FF parameter file used to generate the topology. The first column is the name of the Q-atom type, then follows the Lennard-Jones A_i (or R^*) and B_i (or ε) parameters. Columns four to seven are not used in this case (two parameters for the exponential repulsion function and two LJ parameters for 1-4 interactions). The last column is the atomic mass. The [**change_atoms**] section states that q-atom number one is of type Na in state 1 and type K in state 2.

So, in this example all we have done is to define the relevant LJ parameters for Na^+ and K^+ (Q-atom types for Na and K) as the two different states for our single ion.

Example: Valence bond (EVB) states for a proton transfer reaction

This is an example from the reaction of a protein tyrosine phosphatase where proton transfer from a Cys residue of the enzyme to the doubly negatively charged phosphate group of the substrate (phenylphosphate) is considered. The states representing different bonding arrangements we want to define are schematically drawn in figure 2, where also the topology number of the relevant atoms are given.

The FEP file below describes the two EVB states used for calculating the free energy profile of proton transfer in a particular enzyme [15]. It is beyond the scope here to describe the EVB method in detail, but reviews on this topic are available [5].

Here we want to define the first state with the proton (H) attached to the sulphur atom of the cysteine and the phosphate group doubly charged. In the second state the proton is on a phosphate oxygen and one negative charge is now on the sulphur atom. Here there are changes in both partial atomic charges, vdW parameters, bonds, angles etc. between the two states.

Table 5: FEP file for proton transfer reaction.

Data		**Description**
[FEP]		
states	2	no. of states

Table 5: FEP file for proton transfer reaction.

Data							**Description**
[atoms]							Designate atoms in topology as Q-atoms
1	79						Q-atom no., topology atom no.
2	80						
3	81						
4	1542						
5	1543						
6	1544						
7	1545						
8	1546						
[change_charges]							Assign new charges for each state
1	0.180	0.000					Q-atom no., charges in state 1 & 2
2	-0.450	-1.000					
3	0.270	0.398					
4	0.540	1.230					
5	-0.360	-0.360					
6	-0.860	-0.860					
7	-0.860	-0.860					
8	-0.860	-0.548					
[atom_types]							Define new atom types (LJ parameters, . . .)
!Type	Ai	Bi	Ci	ai	Ai(1-4)	Bi(1-4)	Mass
P	2303.00	59.35	0.0	1.581	2303.00	59.35	30.97
OE	600.00	23.25	70.0	1.581	600.00	23.25	16.00
OD	956.00	23.01	70.0	1.581	956.00	23.01	16.00
H	0.00	0.00	6.5	1.581	0.00	0.00	1.00
C2	2500.00	46.06	0.0	1.581	2500.00	46.06	14.00
SH	2001.57	44.74	165.0	1.581	2001.57	44.74	32.06
S-	2720.00	136.00	165.0	1.581	7200.00	136.00	32.06
[change_atoms]							Assign new atom types to Q-atoms
1	C2	C2					Q-atom no., Q-atom name in states 1 & 2
2	SH	S-					
3	H	H					
4	P	P					
5	OE	OE					
6	OD	OD					
7	OD	OD					
8	OD	OE					
[soft_pairs]							Atom pairs which have C*e$^{(-ar)}$ repulsion
2	3						Q-atom i, j
3	8						

Table 5: FEP file for proton transfer reaction.

Data						**Description**
[excluded_pairs]						Atom pairs to exclude from non-bonded interactions
81	1544	0	1			Atom i, j, exclusion flag for states 1 & 2
81	1545	0	1			
[bond_types]						Define Morse bond types
1	85.0	2.00	1.61			No., D_e, α, b_0
2	120.0	2.00	1.49			
3	84.0	2.00	1.43			
4	110.0	2.00	1.00			
5	94.0	2.00	1.33			
6	112.5	2.00	1.80			
7	100.0	2.00	1.53			
[change_bonds]						Redefine bonds
1542	1546	2	1			Atom i, j, type in state 1 & 2
80	81	5	0			type 0 means no bond
1546	81	0	4			
[angle_types]						Define new angle types
1	95.0	109.6				
2	140.0	120.0				
3	115.0	120.0				
4	110.0	109.6				
5	0.0	0.0				
6	110.0	113.0				
7	95.0	96.0				
[change_angles]						Redefine angles
1544	1542	1546	2	1		Atom i, j, k, type in state 1 & 2
1545	1542	1546	2	1		
1542	1546	81	0	4		Type 0 means no angle
79	80	81	7	0		
[torsion_types]						Define new torsion types
1	0.75	3.0	0.00			Number, force const., mult, delta
2	0.70	3.0	0.00			
[change_torsions]						Redefine torsions
1543	1542	1546	81	0	1	Atom i, j, k, l, type in state 1 & 2
1544	1542	1546	81	0	1	Type 0 means no torsion
1545	1542	1546	81	0	1	
78	79	80	81	2	0	
[angle_couplings]						Define angles to be coupled with Morse bonds

Table 5: FEP file for proton transfer reaction.

Data						**Description**
3	3					Q-angle no., Q-bond no.
4	2					
[torsion_couplings]						Define torsions to be coupled with Morse bonds
1	3					Q-torsion no., Q-bond no.
2	3					
3	3					
4	2					
[off_diagonals]						Define off-diagonal (H_{ij}) functions
1	2	2	8	1.0	0.45	State i, state j, Q-atom 1, Q-atom 2, $A_{i,j}$, $\mu_{i,j}$

In this example we define eight atoms as Q-atoms whose charges, vdW parameters and bonding arrangement will change between the two states (reactant and product state) that we describe by the FEP file. The sections [**FEP**], [**atoms**], [**change_charges**], [**atom_types**] and [**change_atoms**] are used as above, that is, we redefine the charges and vdW parameters of the eight Q-atoms. *e.g.*, atom no. 2, the sulphur, will change its charge from -0.45 to -1.00 and its vdW parameters are changed from Q-atom type SH to S-. In this model of the reaction we will also make use of a non-Lennard-Jones non-bonded potential for certain pairs of atom that make and break bonds as listed in the section [**soft_pairs**]. For these atoms, it is more physical to use an exponential function for the repulsion than the normal $1/r^{12}$ form which causes a too strong repulsion at very short distances. The vdW interaction between these pairs of atoms is given by:

$$V_{soft} = C_i \cdot C_j \cdot e^{(-a_i \cdot a_j \cdot r_{i,j})}$$

where $r_{i,j}$ is the distance between the specific atom pair subjected to this potential. The C's and a's are atom-type specific parameters and the combination rule is geometric as can be seen from the formula. Note the absence of the attractive $1/r^6$ term.

Morse potential parameters for bonds that are broken or formed are given in the [**bond_types**] section. The section [**change_bonds**] lists the bonds, identified by pairs of atoms and the bond parameters to use in each state. If a bond is already defined in the topology then the normal, harmonic potential will be turned off. The absence of a bond is specified by setting the bond type to 0. Bond angles are redefined in an analogous way, but the functional form of the Q-atom angles is harmonic, like the normal angles. Parameters for the new angle types are given under [**angle_types**] and the angles for which the new types should be used are listed in the [**change_angles**] section. Redefining torsions is done in the same way (sections [**torsion_types**] and [**change_torsions**]). No improper torsions are changed in this example.

Angles, torsions and impropers depend on the existence of bonds connecting the atoms

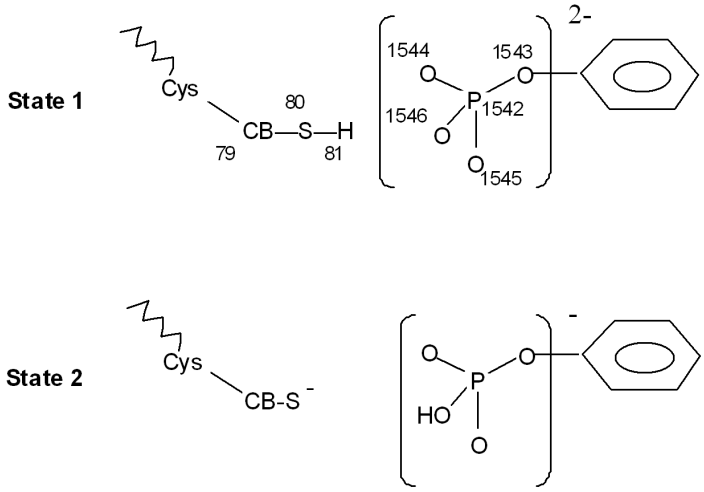

Figure 2: EVB states for a proton transfer reaction.

defining the angle. Angles of all kinds can therefore be coupled to bonds, in which case the angle energy will be scaled by the ratio of the actual value of the Morse bond energy to the dissociation energy [16]. In the example angle 6 (POH) is coupled to bond 3 (OH) and angle 7 (CBSH) to bond 2 (SH), according to the [**angle_couplings**] section. Coupling torsions and impropers (not in the example) work the same way.

Off-diagonal elements of the Hamiltonian are defined in the section [**off_diagonals**]. They are represented by $H_{i,j}=A_{i,j} \cdot e^{(-\mu_{i,j} \cdot r_{k,l})}$ where i and j are the two states involved and $r_{k,l}$ is the distance between a specific pair of atoms k and l. The single record in this example defines mixing of states 1 and 2 ($H_{1,2}$) for q-atoms 2 and 8 with A=1.0 and μ= 0.45.

B.2.6 Monitoring non-bonded interactions

In analysing the details of *e.g.* receptor-ligand interactions, it is useful to define some groups of atoms and calculate the non-bonded interactions between pairs of atom groups. The example FEP file below describes how to use this feature to get the non-bonded energies between the pterinine ring of a dihydrofolate reductase inhibitor and some amino acid side chains and an amide group of a co-factor.

[monitor_groups]							
266	267	268					!GLU 30 COO-
317	318	319	320	321	322		!PHE 34 side chain
1897	1898	1899	1900	1901	1902	1903	!part of MTX pteridine ring 1
1908	1909	1910	1911	1912	1913	1914	!ring 2 of pterindine
1880	1881	1882	1883	1884	1885		!amide of NADPH
[monitor_group_pairs]							
1	3						
2	4						
2	5						

Five groups of atoms are defined, and the interactions between groups 13, 24 and 25 should be calculated. The energies are evaluated separately for different FEP states and presented in the energy summaries in the **qdyn** output. In this example only a single state is defined so the λ-weighted averages are identical to the energies in state 1.

========== Monitoring selected groups of nonbonded interactions ==========

pair	Vwsum	Vwel	Vwvdw	1:Vel	1:Vvdw
1	-58.48	-65.65	7.18	-65.65	7.18
2	-1.72	0.00	-1.72	0.00	-1.72
3	-0.08	0.00	-0.08	0.00	-0.08

where the columns are: atom group pair number, total energy for all states weighted by λ, weighted sum of electrostatic energies, weighted sum of Lennard-Jones energies, electrostatic energy in state 1, Lennard-Jones energy of state 1.

There is a similar feature in Qcalc (see page 70) where one can analyze non-bonded interactions from saved trajectory files.

B.3 Parallel version of Q

Even though computers become faster every year the work that a single computer can do is limited. A single execution of Q will take hours or even days. If a several computers were able to work in parallel with the same job the execution time could be reduced substantially.

The most common way to run a parallel job is to use a computer cluster in which every node has a separate processor and hard drive. The nodes communicate through a fast network switch providing an environment suitable for running parallel program. Q has been parallelised to fit these type of machines.

The part of Q that can be run in parallel is **qdyn** which contain the time demanding conformational sampling. The parallel version is suitable to run on 2 - 20 nodes depending on the size of the problem and the speed of the network.

B.3.1 Performance

To measure how well a parallel program runs there are two quantities. The first one is the speed-up defined as

$$S_p = \frac{\text{sequential time}}{\text{parallel time}} = \frac{T_1^s}{T_p} \tag{1}$$

where T_1^s is the execution time for the best sequential program and T_p is the execution time for the parallel version with p processors. The absolute speedup gives a measure of the improvement achieved by the parallelisation, *i.e.* how many times faster the parallel version is compared to the original.

The second quantity is the efficiency defined as

$$\eta = \frac{\text{sequential time}}{P \times \text{parallel time}} = \frac{T_1^s}{P \times T_p} \tag{2}$$

where P is the number of processors. The efficiency describes how well the total cpu-time is utilised in the parallel version compared to the sequential program.

The speed-up and efficiency of Q was measured using periodic boundary condition for a molecular system with a ion-channel with 1550 atoms solvated by 4514 water molecules. Two series of executions were made with two different cut-offs. The tests were performed at a cluster with IBM SP2-nodes, 160 MHz.

The results of the test series can be seen in figure. 3 and 4. The graphs show the typical behaviour of a parallel program; the more nodes you use the faster executions you get. But at the same time the nodes are utilised less efficient. It is a trad-off between speed and efficiency that is up to the user to decide. When the number of simulations is close to the number of computers it is more efficient to run sequentially; performing 15 simulations on 15 computers is best done by assigning one job per node.

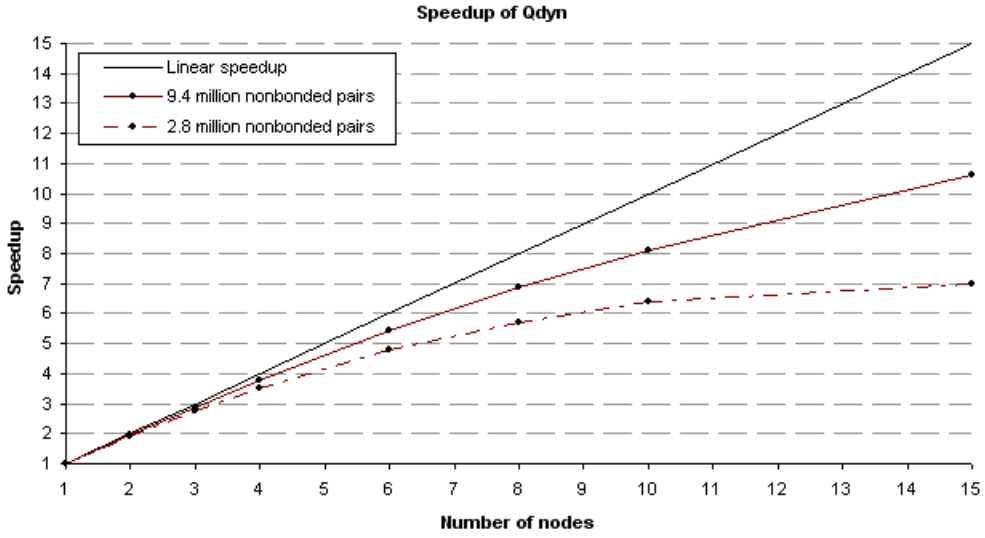

Figure 3: Speedup of the parallel version of **qdyn**.

B.3.2 Running Q on a cluster

Q version 5.0 and higher can be run in parallel on clusters. The parallel version is easy to use. The only requirement is a computer cluster with high bandwidth and a version of Message Passing Interface (MPI) installed. MPI is a standard interface for communication between nodes in a cluster. To run Q on the cluster you need the parallel version of Q, *i.e.* the one that has been compiled with the MPI-flag activated. If you compile the program yourself define the variable USE_MPI to the preprocessor. To check that you have the right version execute the program and confirm that the suffix "_parallel" is added to the version info in the log file. Look at the top of the file where it should read "**qdyn** version

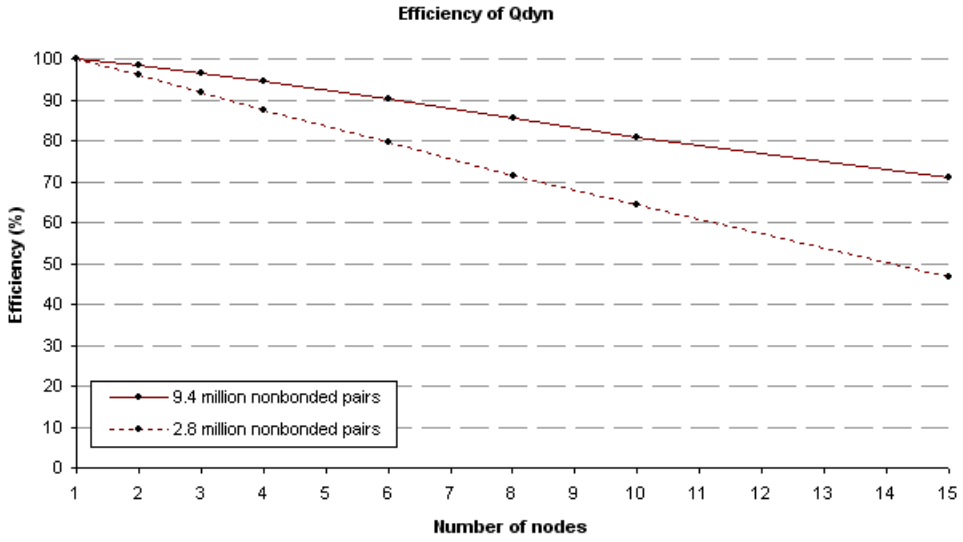

Figure 4: Efficiency of the nodes when running **qdyn** in parallel. The efficiency decreases as the number of nodes increase due to more communication and longer summation of the forces.

X.XX_parallel initialising....". The parallel version can be executed on a single node but still requires MPI installed.

When submitting jobs to a cluster Q will automatically detect how many nodes that are available. Thus no special input to Q is required about how many nodes are being used. Make sure you run on dedicated nodes, *i.e.* you have exclusive access to the nodes, and that no other job is running on the nodes. If the nodes are not dedicated entirely to Q the parallel execution become meaningless as a result of the synchronisation between the nodes in each time step. Consult your system administrator on how to commit jobs to a specified number of dedicated nodes.

B.4 Analysis of results

B.4.1 Analysing structures from the simulation

Qprep is also used after the simulation to convert the binary trajectory and restart coordinate files generated by **qdyn** to PDB or mol2 files suitable for viewing in a molecular graphics program.

Making PDB or mol2 files from restart or trajectory files

Use the following steps to generate viewable files from individual "snapshot" structures:

1. Load the topology file using the **readtop** command. The fragment library files used to generate the topology will be loaded automatically, if available. Otherwise load the libraries using **readlib**.

2a. Load the binary coordinate file using **readx**.

2b. Open a trajectory file with the **trajectory** command. You will be prompted if you want to use the atom mask from the trajectory so that only atoms in the trajectory will appear in structure files written. Read coordinates from the trajectory file with **readframe**

3. If you want to select specific atoms to include in the structure files to be written, use the **mask** command . First enter mask none to clear the current mask, then add atoms to the mask using the syntax described in the section Atom mask on page 42.

4a. Write a PDB file using the **writepdb** command. It may be written with or without gap markers.

4b. Write a mol2 file using the **writemol2** command.

5. Repeat steps 2 and 4 to convert more files. To read the next frame from a trajectory use the **readnext** command and then go to step 4. Note that CONECT records in PDB files (defining atomic connectivity) will only be generated for fragments whose library entries include PDBtype HETATM in their info section.

Trajectory animation

There are a number of programs for visualising and analysing MD trajectories, *e.g.* Visual Molecular Dynamics, VMD [18, 19] and gOpenMol [20, 21] which can read the DCD format trajectories generated by **qdyn**. To create a PDB file with the same set of atoms as in the trajectory, as required by the visualisation programs, execute the steps below in Qprep:

1. Load the topology file using the **readtop** command. The fragment library files used to generate the topology will be loaded automatically, if available. Otherwise load the libraries using **readlib**.

2. Open a trajectory file with the **trajectory** command. You want to use the atom mask from the trajectory (answer y for yes at the prompt).

3. If you want to use another coordinate set than that of the topology for your PDB file, use **readframe** or **readx** as described above.

4. Write a PDB file using the **writepdb** command. Don't include gap markers.

B.4.2 Free energy calculation using Qfep

Performing free energy perturbation (FEP) calculations with Q involves running a set of consecutive input files which have the mapping parameter vector λ ranging in a desired way (usually [1, 0] to [0, 1] for two states). Qfep is a program which reads the energy files generated by **qdyn** and calculates the total change in free energy for the complete perturbation from state A (ε_1) to state B (ε_2). The difference in free energy between the two states is calculated by Zwanzig's formula:

Table 6: Qfep input file for FEP/EVB evaluation

Line	**Data**	**Description**
1	11	Number of energy files
2	2 0	Number of states, number of predefined off-diagonal elements (from .fep-file, 0 means redefine)
3	0.596 100	kT, number of points to skip in each energy file
4	40	Number of gap bins
5	20	Minimum number of points per bin
6	12.3	Energy shift $\Delta\alpha_{ij}$ (for states$\neq$1)
7	1	Number of off-diagonal elements $\neq$0
7.1	1 2 18.1 0.32 0.0 2.0	i, j, A_{ij}, μ_{ij}, η_{ij}, r_{0ij}
8	1 -1	Linear combination of states defining the reaction coordinate ($\varepsilon_1 - \varepsilon_2$).
9	md_00.ene	List of energy files
10...	md_01.ene	
...19	md_10.ene	

$$\Delta G = \sum \Delta g = \sum -R \cdot T \cdot \ln \left\langle e^{-\left(\frac{\Delta V_{eff}}{R \cdot T}\right)} \right\rangle_A \tag{3}$$

where $V_{eff} = \lambda_1 \cdot \varepsilon_1 + \lambda_2 \cdot \epsilon_2$, $\Sigma\lambda_n = 1$. ΔV_{eff} is the difference in V_{eff} between two adjacent perturbation steps.

The program returns a list containing average energies and lambda values for each file. After that, the free energy change between each perturbation step (file) is summarised. The change is calculated relative to both the previous and the following perturbation step (dGf and dGr for forward and reverse way respectively). The accumulated sum of the energy changes between ε_1 to ε_2 is also given (sum(dGf) and sum(dGr)), as well as the average accumulated change calculated from the forward and reverse way $\langle$dG$\rangle$.

Qfep also calculates free energy functions, or potentials of mean force, by the perturbation formula:

$$\Delta G(X_m) = \Delta G(\lambda_i) - R \cdot T \cdot \ln \left\langle e^{-\left(\frac{E_g(X_m) - V_i(X_m)}{R \cdot T}\right)} \right\rangle_i \tag{4}$$

The reaction coordinate X is defined as the energy gap between the states $X = \Delta V = \varepsilon_1 - \varepsilon_2$ and is divided into intervals X_m (bins). The first term in the above equation represents the free energy difference between the initial state ε_1 and the mapping potential V_i:

$$\Delta G\left(\lambda_i\right) = -R \cdot T \cdot \ln \sum_{n=0}^{i-1} \left\langle e^{-\left(\frac{V_{n+1} - V_n}{R \cdot T}\right)} \right\rangle_n \tag{5}$$

The second term represents the free energy difference between the mapping potential V_i and the ground state potential E_g. The MD average in this term is only taken over those

configurations where X belongs to X_m. E_g is the solution to the secular determinant:

$$\begin{vmatrix} \varepsilon_1 - E_g & \cdots & H_{1n} \\ \vdots & \ddots & \vdots \\ H_{n1} & \cdots & \varepsilon_n - E_g \end{vmatrix} = 0 \tag{6}$$

For a two-state representation the solution becomes:

$$E_g = \tfrac{1}{2} \cdot (\varepsilon_1 + \varepsilon_2) - \tfrac{1}{2} \cdot \sqrt{(\varepsilon_1 + \varepsilon_2)^2 + 4 \cdot H_{12}^2} \tag{7}$$

H_{ij} is the off-diagonal matrix element representing the quantum mechanical coupling of the states. This coupling is zero for normal FEP calculations. $H_{ij} \neq 0$ results in mixing of states i and j, which is desired when calculating reaction free energy profiles for reactions represented by the empirical valence bond (EVB) model. In Qfep the off-diagonal element is a function of the form:

$$H_{ij} = A_{ij} \cdot (e^{-(\mu(r_{ij}-r_0)+\eta(r_{ij}-r_0)^2)}) \tag{8}$$

where r_{ij} is the measured distance between the reacting atoms. By choosing μ and η differently, H_{ij} can either be a constant, an exponential function or a gaussian function.

The EVB method allows calibration of simulated reference reactions to experimental data obtained from gas-phase or solution experiments. The two EVB parameters H_{ij} (mostly A_{ij}) and $\Delta\alpha_{ij}$ are varied until the calculated profile and the experimental data coincide. $\Delta\alpha_{ij}$ is a constant energy shift between the states that represents their difference in heat of formation, which is not included in the force field. Generalised, the $\Delta\alpha_{ij}$ parameter determines the ΔG° level and H_{ij} regulates the degree of mixing of the states at the transition state *i.e.* the $\Delta G^\ddagger$ level.

The energies describing the FEP functions and the reaction free energy profile are summarised in the last table generated by Qfep. Note that each bin has contributions from several different values of λ. Likewise, each value of λ contributes to the sampling of several different bins.

It is possible to handle more than two valence bond states in **qdyn** and Qfep, however sampling and calibration may be a difficult task for more than two states.

B.4.3 Reaction free energy averaging using avefep

In the reaction free energy calculations described above, several mapping potentials (different λ's) will sample each interval X_m. The resulting values for the potential of mean force have to be averaged to obtain the final values of ΔG associated with each interval X_m. Avefep calculates the average values (weighted by sample size) of ΔG corresponding to each interval from Qfep output data.

Input is an output-file created by Qfep.

Figure 5: Example Qfep output file

```
...file opened, and lambdas: md_00.ene   0.00 1.00
 Number of points:         1000
state   EQtot   EQbond  EQang   EQtor   EQimp   EQel    EQvdW
-------------------------------------------------------------
   1 -1087.71 -578.19    7.68    0.63    0.00 -541.46   22.70
   2 -1145.21 -586.25   31.48    1.16    0.00 -603.21   10.68

state   Eel_qq  EvdW_qq Eel_qp  EvdW_qp Eel_qw  EvdW_qw Eqrstr
-------------------------------------------------------------
   1   112.02    0.57  -18.78   -1.97 -634.70   24.10    0.00
   2     0.00   -0.30   -5.32   -2.51 -597.89   13.50    0.00
etc...
lambda(1) dGf   sum(dGf)    dGr   sum(dGr)    <dG>
-------------------------------------------------
  0.00     0.00     0.00    -5.71  -284.20     0.00
  0.01     5.64     5.64    -5.54  -278.49     5.67
  0.02     5.63    11.27    -5.45  -272.95    11.26
  0.03     5.48    16.74    -5.28  -267.50    16.72
  0.04     5.38    22.12   -10.41  -262.22    22.05
  0.06    10.37    32.49   -10.07  -251.80    32.44
etc...
Min energy-gap is:   -292.238384870069
Max energy-gap is:    330.166319351831

Lambda1 Ibin Energy gap  dGa    dGb      dGg    # pts
-----------------------------------------------------
 0.00    1  -284.46   -31.28   247.20   -47.72     65
 0.00    2  -268.90   -31.28   231.53   -48.50    200
 0.00    3  -253.34   -31.28   216.31   -49.35    231
 0.00    4  -237.78   -31.28   201.25   -50.31     88
 0.01    2  -268.90   -31.40   231.23   -48.73     79
 0.01    3  -253.34   -31.26   216.16   -49.42    303
 0.01    4  -237.78   -31.13   201.59   -50.14    181
 0.01    5  -222.22   -30.99   189.55   -51.00     25
etc...
```

Output is a table with the average ΔG corresponding to the reaction co-ordinate X.

Usage: perl avefep.pl < qfep.out > avefep.out

B.4.4 Simple energy averaging using Qave

Qave calculates average energies and RMS deviations from .log-files. The program uses one line of input:

perl Qave.pl *[-s nskip]* files

The value *nskip* is the number of steps to be discarded at the beginning of the log-file. Default is zero. A number of log-files can be listed after one another and the average energies calculated from all files will be returned.

B.4.5 Trajectory averaging using Qavetr

Average structures from trajectory files are generated by the program Qavetr. Execute the program by typing:

col.	description
1	Bin number
2	Energy gap, X
3	Average ΔG
4	Average ΔG, lowest point at 0.
5	Number of points in bin

Qavetr

The program will ask for the trajectory file to be processed. More than one trajectory file may be used in the averaging. Enter the files one at a time and end with a blank line. Enter the name of the output file that will contain the coordinates of the average structure. The output file is binary but can be converted to a pdb-file by using the program Qprep. Root mean square coordinate deviation is given in Å.

B.5 Scoring

Three scoring functions are implemented in QCalc: X-Score[32], ChemScore[33] and PMF-Score[34]. X-Score and ChemScore are empiricial whilst PMF-Score is knowledge based.

All scoring functions require the topology to be loaded and the correct mask be specified. The initial topology (with coordinates from the .top-file) can be scored to verify atom typing.

Both trajectory and restart files can be scored. The following options are avaliable in QCalc when specifying trajectory or restart files:

- adding *,frames=every n* means calculations will only be performed on every n:th frame.
- adding *,frames=n-m* means calculations will only be performed on frames n to m.
- specifying *mean* instead of a file name calculates the mean of all frames processed since start or since the last time "mean" was specified.

The input requested is similar for all three functions. To avoid confusion, examples of typical inputs will be given.

B.5.1 X-Score

Input Example input is presented below.

Prompt	Input
Topology file:	c:/peter/data/P450/adm/adm.top
Qcalc>	xscore
Mask:	residue 1 407
Mask	.
Score initial topology? (yes/no)	yes
Q-atom (FEP) file:	c:/peter/data/P450/adm/lig.fep
Cofactor (. or EOL terminates):	restype=HEM
Cofactor (. or EOL terminates):	.
FF translation key:	qoplsaa
Scoring parameters:	xscore_default.input
Qcalc>	go
Enter names of coordinate or restart files	c:/peter/data/.../md.dcd,frames=every 5 mean

Cofactors X-Score uses different typing schemes to assign atoms types to protein and ligands atoms. If needed, atoms in parts of the protein can be typed using the ligand atom-typing procedure by defining a cofactor. This is useful if the protein has some special residue, like HEM in P450, that is not defined in the X-Score residue library (file RESIDUE_DEF_XTOOL.dat). Ligand atoms are typed on the individual atom level, in contrast to the residue level for protein atoms, using data in file ATOM_DEF_XTOOL.dat. Cofactor definitions are made on separate lines and has the form: restype=RES, where RES is the residue name, e.g. HEM. All atoms, regardless of their proximity to each other, in residues named RES will be included in the cofactor RES and typed as if they were ligand atoms (though in every other respect they are considered as part of the protein). Any number of cofactors can be defined.

Force field A force field translation key has to be given to allow for the translation of atom types according to the Q convention to types according to the Sybyl convention. The translation tables are in the file ATOM_TYPE_CONVERSIONS.dat (shared with PMF-Score). A different translation file can be specified in the input file (see below).

Parameters Scoring parameters, output specifications and data files are specified in an input file. Default parameters can be used by specifying *default* when asked for scoring parameters. In that case the following parameters and filenames are used:

```
SHOW_ABS          'NO'    ! Show binding score for each atom?
SHOW_TOTAL        'YES'   ! Show total binding score?
SHOW_LIGAND       'YES'   ! Show ligand atoms?
SHOW_PROTEIN      'NO'    ! Show protein atoms?
SHOW_COFACTOR     'YES'   ! Show cofactor atoms?
APPLY_HPSCORE     'YES'   ! Use hydrophobic contact algorithm?
APPLY_HMSCORE     'YES'   ! Use hydrophobic matching algorithm?
APPLY_HSSCORE     'YES'   ! Use hydrophobic surface algorithm?
```

RESIDUE_DEFINITIONS	residue_def_xtool.dat
ATOM_DEFINITIONS	atom_def_xtool.dat
LOGP_DEFINITIONS	atom_def_xlogp.dat
SURFACE_DEFINITIONS	surface_def_xtool.dat
ATOM_TRANSLATIONS	atom_type_conversions.dat

Applying more than one hydrophobic algorithm results in a consensus score. Default scoring coefficients are as reported in [32].

Output When SHOW_LIGAND and/or SHOW_PROTEIN is specified, a list of ligand and/or protein atoms is displayed, showing the atom type, residue, atomic properties, neighbouring atoms and bonds for each atom. In addition, a list of bonds and aromatic rings detected are output.

When scoring, the contribution from each term (van der Waals (VDW), H-bonding (HB), hydrophobic contact (HP), matching (HM) and surface (HS) and rotational(RT)) is displayed along with the total score. If SHOW_ABS was specified the contributions for every ligand atom is displayed. Atomic binding score is always displayed when scoring the initial configuration.

X-Score results are in pK_d units.

Data files The format of the data files is further explained in the respective files.

B.5.2 ChemScore

Input Example input is presented below.

Prompt	Input
Topology file:	c:/peter/data/P450/adm/adm.top
Qcalc>	chemscore
Mask:	residue 1 407
Mask	.
Score initial topology? (yes/no)	yes
Q-atom (FEP) file:	c:/peter/data/P450/adm/lig.fep
Parameter file:	c:/peter/data/ff/chemscore_oplsaa.prm
Qcalc>	go
Enter names of coordinate or restart files	c:/peter/data/.../md.dcd,frames=every 5 mean

Parameter file ChemScore reads all atom parameters from a single parameter file, though there are different files for different force fields. The parameter file defines the atomic properties of all atom types.

Output Prior to scoring, ChemScore outputs atom type and bond information for all ligand atoms, as well as information about rings detected. For every frame, the contribution from each term (H-bonds, metal contacts, lipophilic contacts and frozen rotatable bonds) is displayed along with the total score.

ChemScore results are in kJ/mol. A negative score means negative energy.

B.5.3 PMF-Score

Input Example input is presented below.

Prompt	Input
Topology file:	c:/peter/data/P450/adm/adm.top
Qcalc>	chemscore
Mask:	residue 1 407
Mask	.
Score initial topology? (yes/no)	yes
Q-atom (FEP) file:	c:/peter/data/P450/adm/lig.fep
Force field translation key:	qoplsaa
Scoring parameters:	pmfscore_default.input
Qcalc>	go
Enter names of coordinate or restart files	c:/peter/data/.../md.dcd,frames=every 5 mean

Presently, all atoms defined as solvent atoms are ignored. Critical water molecules should be defined as part of the protein or they will be excluded.

Force field As for X-Score, a force field translation key has to be given to allow for the translation of Q atom types to Sybyl atom types. The Sybyl type derived is only used to determine the hybridization of carbon and nitrogen atoms.

Scoring parameters Output options, data files and the maximum ring size considered are defined in an input file. The output options are similar to those in X-Score. The maximum ring size parameter determines the number of steps the ring finding algorithm will take in every search direction. Too low a setting will prevent the algorithm from finding all rings. Too high a setting will increase the time required for the ring search.

Output When SHOW_LIGAND and/or SHOW_PROTEIN is specified, a list of ligand and/or protein atoms is displayed, showing the atom type, residue, atomic properties, neighbouring atoms and bonds for each atom. In addition, a list of rings detected are output. If specified, bonds are also output. Atomic binding score is displayed only when scoring the initial configuration (topology).

It is safe to consider PMF-Score results as rankings where a more negative score means better binding. For details about converting PMF-Score to free energy of binding, see [34].

B.6 Useful tips

- To run FEP simulations of a ligand in water and bound to a protein using the same FEP file, use the **offset_name** keyword in the [**FEP**] section of the FEP file to instead of renumbering all the atoms!
- Make a separate library file for your new molecules and leave the amino acid library unchanged. Load both library files in Qprep!
- It is possible to add parameters to the parameter file without restarting Qprep. Just type maketop and the updated file will be used!
- For FEP simulations involving dummy atoms, the daring user might consider ignoring some Qprep warnings about missing parameters all of those interactions are to be redefined in a FEP file. It is possible, but in general not advisable, to write a topology file with missing parameters and to use it in **qdyn**.
- Use build rules in your fragment library entries to control the positioning of hydrogens.
- Improve scoring accuracy by averaging over e.g. every 10:th frame of a short equilibration trajectory file!

C TUTORIALS

C.1 Binding affinity from LIE simulations

The example here is the binding of stearate to a muscular fatty-acid binding protein. We have used the Q version of the GROMOS87 forcefield for the simulations.

C.1.1 Editing the PDB file

The structure of the M-FABP complex with stearate, PDB-idcode 1HMT was downloaded. The PDB-file needs some editing before use, first you have to remove some of the crystal waters, if any. In the 1HMT-file, 17 waters were saved, having an important role in the binding with the ligand or in other interactions. To decide which waters to save, pick an atom in the ligand to be the centre of your system and choose how large simulation sphere you are going to use. Here, a sphere of 18.0 Å radius has been used. Then keep the waters inside the sphere that seem to bee involved in any interactions and that lie inside the protein. We have deleted the waters by hand in a molecular viewer program and then saving only the lines holding the coordinates. The part left from the original pdb-file is the coordinates of the protein, ligand and some waters. But it takes some more editing. All lines that are blank or say TER also have to be removed. There has to be a line saying GAP between the different molecules. All the cysteines that are connected through sulphur bridges should be renamed CSS.

Note also that hydrogen atoms need not to be present in the PDB file, they will be added by the Qprep program.

C.1.2 Modelling ionic groups of the protein

Be aware that the default model of the charged amino acid residues (ASP, GLU, ARG, LYS) in the Q-GROMOS87 fragment library have the protonation state of the ionic form, but the net charge replaced by a dipole similar to that of neutral form. The corresponding charged side chains are described by library entries AS-, GL-, AR+ and LY+, respectively. Below, we refer to the process of renaming *e.g.* an ASP residue to AS- as "turning on" the charge of that side chain.

Now it is time to decide which of the charged amino acids that should be "turned on". You can use the same rules here as in choosing which waters to keep. Amino acids near the ligand, creating an salt bridge or interacting in any other way in the function of the protein and which lie inside the 18.0 Å sphere should be charged. Residues closer than about 3-5 Å from the boundary should be neutral unless they form an ion pair with a more central group. In a case like this, when the ligand has an ionic group, is it important to make the protein net neutral. In this example, amino acids 17, 72, 76, 77, 78, 106 and 126 were charged. This is most easily done in the program proq (see page 70). After loading your pdb-file (*load* filename.pdb), one good thing to do is make a subset centre (*centre* name coordinates), of the centre atom you chose before. From this subset, you can measure distances to different residues (*dist* subset subset), like the ones you want to charge, and the water molecules. You turn on the amino acid charges by the command on followed by the residue number (*on* nr). Remember to save the new pdb-file before exiting proq (*save* filename.pdb). There are many other useful functions in the proq program, type *help* and you will get a list of them.

C.1.3 Writing the library file

The next step is to write a library file for the ligand. This is easiest done by editing an old library file. You can also get a lot of help from looking at the amino acid library file (*e.g.* Qgrm87.lib). In the lib file, all the atoms of the ligand, the bonds and the charge groups are listed. For each atom, you need to specify name, type and charge. Make sure the charges in the complex file add up to the charge of the ligand file. All the different types of atoms are listed in the parameter file (Qgrm87j.prm). For the charges, one can sometimes compare with an amino acid from the amino acid library file. For stearate, the lib file is stearate.lib. Also make sure you name the atoms the same way as in the pdb file. Since the ligand will not be connected to any other fragment, the head and tail connections can be omitted.

You also need a pdb file for the ligand, copy the relevant lines from the PDB file of the complex to a separate file.

C.1.4 Qprep

Now it is time to make the topology files. They contain all the information about the system and are used in the **qdyn** simulation. This is done in Qprep, where the sulphur bridges also are created. You have to make two topology files, one for the ligand in protein simulation and one for the ligand without the protein simulation. In Qprep, you start

by reading the library files and the pdb files (*readlib* filename.lib, *readpdb* filename.pdb). In the ligand-protein topology, both the amino acid library and the ligand library files has to be read, and the sulphur bridges created. This is done by the command addbond. (*addbond* atomnumber atomnumber). For the atom numbers of the sulphurs, you can list the atoms in the cysteins (*listres* residuenumber). Note that you cannot use the numbers from the PDB file since atoms will be renumbered as hydrogens are added. After this you select boundary and solvate the system. Then it is time to create the topology, *maketop*. If there are any parameters missing, Qprep will tell you here. To create new parameters, edit the parameter file in a way that you can see the changes made. You can always write comments in the file after a "!". After saving the modified parameter file, all you need to do is *maketop* again. To write the topology file, use the command *writetop.*

There are many other useful applications to Qprep, among other things you can list the high energy bonds, angles, torsions and impropers by the commands *checkbond* energylevel, *checkangs* energylevel and so on. If you get a too high bond, angle or torsion energy, perhaps you have connected the sulphur bridges wrongly or forgotten a GAP between two molecules. If an improper has a very high energy, it might have the wrong sign (e.q. 180 instead of -180 degrees), use the command changeimp to redefine them automatically (*changeimp* 2 energylevel). After using changeimp, you need to write the topology again. You may also want to make a new PDB file, use *writepdb,* with atoms renumbered and hydrogens added. By typing *help*, Qprep lists all the commands with a small explanation.

C.1.5 FEP files

In this example the only thing specified in the FEP files are the Q-atoms, that is, the ligand. In the simulation without the protein, this is simply all the atoms, but in the protein-ligand simulation, you have to find the atom numbers of the ligand in the new pdb file. There are a lot of other things that can be specified in the FEP file, but none of those functions are used in this example.

C.1.6 Creating input files

The input file controls the simulation in **qdyn**. It contains information on how many steps, how long steps, what temperature, which topology to use and a lot of other things. In this example, the data collection phase was split into five identical, consecutive steps to make it easier to restart after an interruption. This gives five input files to each of the two simulations and another 6 for the equilibration of the ligand-protein complex. An input file is easiest created by editing an old input file. The things that need to be specified to a specific simulation are the centre of the simulation sphere and water sphere, the topology- and FEP files, and restraints. The coordinates of the water- and simulation sphere should be the same, coming from the atom in the ligand that you picked earlier.

In this example, the equilibration warms up the system, starting with 0 degrees gradually raising the temperature to 300 degrees. All the heavy atoms, including those of the ligand, are restrained during this equilibration. When the system is equilibrated, 5x50 000 simulation steps of 1 fs are taken for both of the simulations. All the files except the initial one are restarted from the coordinates and velocities of the previous step. When a new

temperature is given, you also need to give a random seed. When the temperature is the same as in the previous file, set the random seed to zero.

The coordinates of the water sphere must be specified in the first input file. In this example, a co-ordinate file with randomly oriented water molecules on a grid, was used.

C.1.7 Using qdyn

To start the simulation, write: **qdyn** filename.inp > filename.log, to use a specific input file and write the output to a log file. When many input files are used it is much easier to write all the commands to a command file and then run that.

C.1.8 Evaluating the simulation

By using the script lsextr (*lsextr* md0*.log > filename.txt), the van der Waals and electrostatic energies are respectively extracted from the log files. It is a good idea to plot these energies, *e.g.* by using gnuplot, to see if there are any large changes in the energies throughout the simulation, see figure 6.

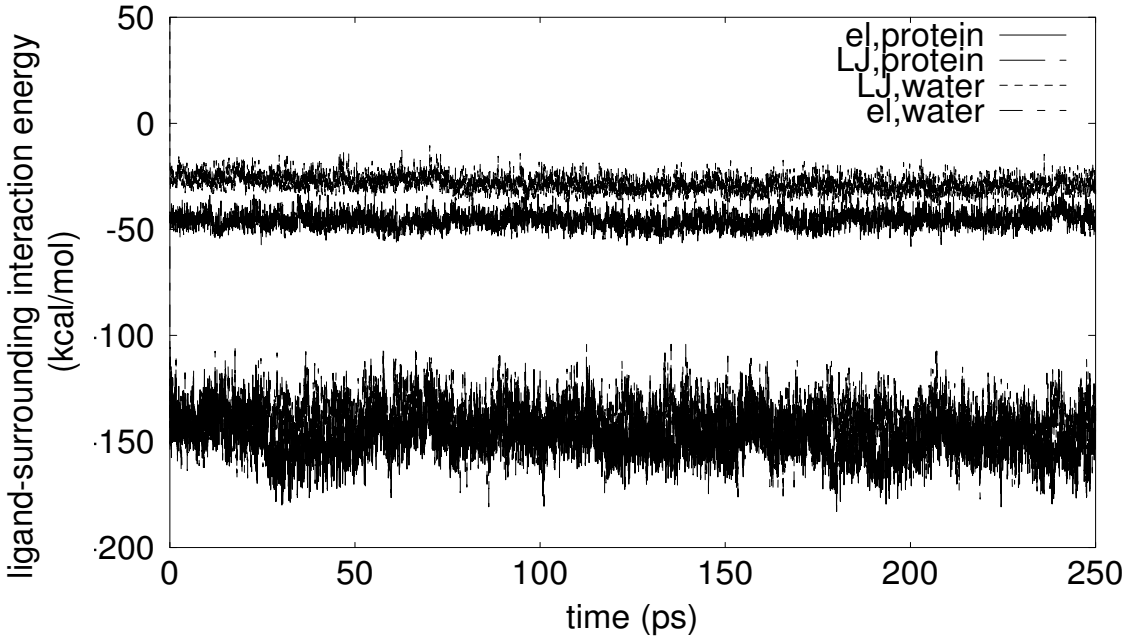

Figure 6: Energies

Viewing the structures after the different parts of the simulation is a very important part of the evaluation of the simulations. To make pdb files of the restart files use Qprep. Read the topology file, then read the restart file (*readx* md0#.re) and write the new pdb file.

To get averages of the different energies, a program called tstart (*tstart* q filename.txt # # #), was used. Tstart calculates the overall average and also an average where you can split the energies in two, each giving the same average, skipping the first values. The numbers

in the command are different choices you can make where the first is either 1 or 2, van der Waal or electrostatic energies. With the second you can choose where to start, zero meaning the beginning, the third how many rows to read, zero meaning all.

The next thing to do is to calculate the electrostatic free energy contribution from the ligand's interaction with ionic groups of the protein that were neglected (not "turned on") during the simulation. We approximate this free energy using Coulomb's law with $\varepsilon = 80$. Run proq, load the pdb file and read the ligand library used in Qprep. The command onoff (*onoff #-##*) switches the charges, charging the uncharged and neutralising the charged. Then calculate the electrostatic energy, ΔG_{onoff}, (*repel* ligand *charged*). This energy should be small, less than 1 kcal/mol. If not, some amino acids that should be turned on probably were not.

Then you can calculate the binding free energy, using the LIE formula:

$$\Delta G_{bind} = \alpha \left(\langle V_{l-s}^{LJ} \rangle_{bound} - \langle V_{l-s}^{LJ} \rangle_{free} \right) + \beta \left(\left\langle V_{l-s}^{el} \right\rangle_{bound} - \left\langle V_{l-s}^{el} \right\rangle_{free} \right) + \Delta G_{onoff} \quad (9)$$

In this example, we use $\beta = 0.5$ (no deviations from electrostatic linear response for a charged ligand) and $\alpha = 0.181$ (from previous calibration using GROMOS87). This gives a ΔG_{bind} = -8.0 kcal/mol, which is close to the experimentally determined value of ΔG_{bind} = -7.9 kcal/mol.

The β-value varies with the number of OH - groups on the ligand, when using GROMOS87 for ligands with no ionic groups, β should be selected from a set of values accordingly to the composition of the ligand (number of OH-groups).[8]

D REFERENCE GUIDE

D.1 Program modules

Q is build from Fortran90 modules, which are combined in different sets in the Q programs, as shown in figure 7. This makes it easier to maintain the software. It also makes it rather straight-forward for users with experience in programming to create their own special-purpose programs by re-using *e.g.* the trajectory and topology modules.

The figure does not show the dependence of some modules on others.

D.2 Force field reference information

Q is not associated with any particular force field. The force fields are defined in parameter files, separate from the program and the choice of force field is thus simply a matter of which parameter file to use. Any force field could be used with the program, as long as it shares the common functional form of eq. 10.

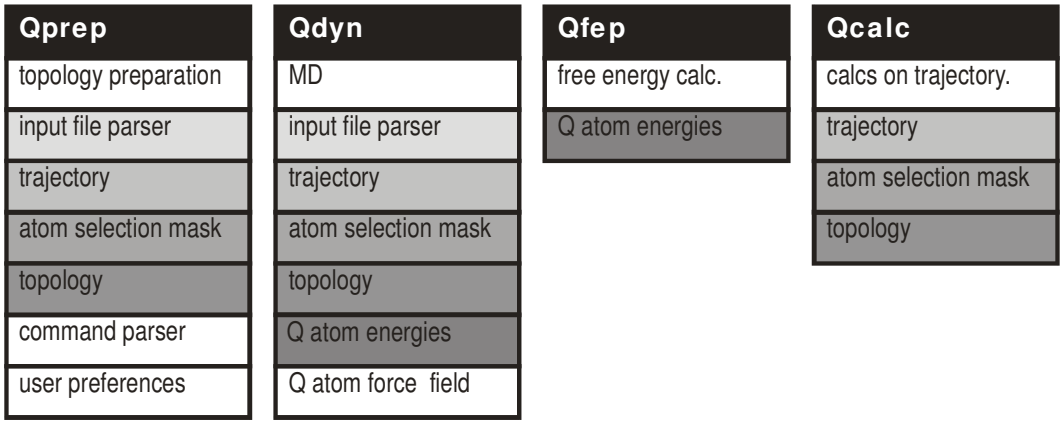

Figure 7: The modules that build up Q.

$$V_{pot} = \sum_{bonds} \frac{1}{2} k_b \cdot (r - r_0)^2 + \sum_{angles} \frac{1}{2} k_\theta \, (\theta - \theta_0)^2 + \sum_{dihedrals} K_\varphi \cdot (1 + \cos(n \cdot \varphi - \delta)) + \sum_{\substack{improper \\ dihedrals}} \frac{1}{2} k_\xi \, (\xi - \xi_0)^2 + \sum_{\substack{atom \\ pairs\ i,j}} \frac{1}{4 \cdot \pi \cdot \varepsilon_0} \cdot q_i \cdot q_j \cdot r_{ij}^{-1} + A_{ij} \cdot r_{ij}^{-12} - B_{ij} \cdot r_{ij}^{-6} \quad (10)$$

where V_{pot} is the total potential energy, k_b is a bond stretching force constant, r is the distance between two bonded atoms, r_0 is the reference bond length, k_θ is an angle bending force constant, θ the angle between two bonds, θ_0 is the reference angle, k_φ is a force constant for rotation around a dihedral angle, n is the multiplicity (number of minima per full turn) of the dihedral angle φ, δ is the phase shift (δ/n gives the location of first maximum), k_ξ is an out-of-plane bending force constant for the improper dihedral angle ξ with reference angle ξ_0, q_i and q_j are the partial charges of atoms i and j separated by the distance r_{ij}. A_{ij} and B_{ij} are the geometric Lennard-Jones parameters for the interaction between atoms i and j. The Lennard-Jones parameters are defined per atom type as A_i and B_i and are combined using either of the two standard rules to determine the effective interaction parameters. The geometric rule is simply: $A_{ij} = A_i \cdot A_j$ and $B_{ij} = B_i \cdot B_j$ where $A_i = A_{ii}^{1/2}$ and $B_i = B_{ii}^{1/2}$. Some force fields use the form:

$$\varepsilon_{ij} \cdot \left(\left(\frac{R_{ij}^*}{r_{ij}} \right)^{12} - 2 \cdot \left(\frac{R_{ij}^*}{r_{ij}} \right)^{6} \right)$$

for the 6-12 Lennard-Jones potential. In this case the atom type-parameters ε_i, ε_j, R_i^* and R_j^* are combined using the rules: $\varepsilon_{ij} = (\varepsilon_i \cdot \varepsilon_j)^{1/2}$and $R_{ij}^* = R_i^* + R_j^*$. Several Fourier components of the dihedral terms, with different K_φ, n and δ, can be added for the same dihedral angle to allow a more accurate modelling of the barriers for rotation. An alternative form of the improper dihedral potential using trigonometric functions just as for normal dihedrals is also implemented.

The molecular fragments, *e.g.* amino acid residues, defined in the force fields are divided into

charge groups which are groups of atoms whose partial charges add up to an integer. Cut-off of non-bonded interactions is then done based on these groups, *i.e.* either all pairwise interactions between the two groups are evaluated, or none. The average size of the charge groups varies between force fields, from a few atoms to entire residues. Some force fields designate a "switching centre" in each charge group and performs cut-off only based on the distance between the switching centres, while others include all interactions between two groups if any pair of atoms is within the cut-off radius.

Some properties of the force fields available for Q are given in table 7. Please note that these are our translations of the force field and we cannot guarantee 100% identity with the original.

Table 7: Force fields available in Q.

Force field[1]	**CH$_n$ groups**[2]	**LJ type**[3]	**Impropers**[4]	**Charge groups**[5]	**Cut-off**[6]	**Atom types**[7]	**Ref**
Amber95	all-atom	ε_{ij},R$_{ij}$	periodic	residues	any	48	[22]
Amber/OPLS	extended	A$_{ij}$,B$_{ij}$	periodic	$\leq$ 11 atoms	switching	39	[23]
CHARMM v.22	all-atom	ε_{ij},R$^*_{ij}$	harmonic	$\leq$ 13 atoms	switching	186	[24]
GROMOS87	extended	A$_{ij}$,B$_{ij}$	harmonic	$\leq$ 10 atoms	switching	28	[25, 26]
GROMOS96	extended	A$_{ij}$,B$_{ij}$	harmonic	$\leq$ 10 atoms	switching	28	[27]
OPLS-AA	all-atom	A$_{ij}$,B$_{ij}$	periodic	$\leq$ 15 atoms	switching	35	[23]

The file names of the different fragment libraries and parameter files are given in Table 8.

Table 8: Force field files.

Force field	**Library file**	**FF parameter file**
Amber95	Qamber95.lib	Qamber95.prm
Amber/OPLS	Qopls.lib	Qopls.prm
CHARMM v.22	Qcharmm22.lib	Qcharmm22.prm
GROMOS87	Qgrm87.lib	Qgrm87.prm
GROMOS96	Qgrm96.lib	Qgrm96.prm
OPLSAA	Qoplsaa.lib	Qoplsaa.prm

[1] Our implementation of the named force field.

[2] Hydrogen atoms on aliphatic carbons may either be explicitly treated (all atom) or modelled as an extended atom.

[3] The Lennard-Jones potential can be written either as $\frac{A_{ij}}{r^{12}} - \frac{B_{ij}}{r^6}$ or $\varepsilon_{ij} \cdot \left(\left(\frac{R^*_{ij}}{r_{ij}} \right)^{12} - 2 \cdot \left(\frac{R^*_{ij}}{r_{ij}} \right)^6 \right)$, using the geometric or arithmetic rules, respectively, to combine parameters for pairs of atom types. Treatment of 1-4 interactions (LJ and electrostatic) is specific for each force field.

[4] Improper dihedrals can be modelled either with harmonic potentials or with a periodic function like ordinary dihedrals.

[5] Typical number of atoms in a charge group.

[6] Cut-offs are always applied to whole charge groups and are based either on the distance between designated switching atoms or on the smallest distance between any pair of atoms in two charge groups.

[7] The number of different atom types defined in the Q implementation of the force field.

D.3 Topology preparation reference

D.3.1 Coordinate files for input into Qprep

Atomic coordinates are entered into Qprep as PDB files. The PBD file must conform to some rules to be accepted by Qprep (if not specified use the PDB standard):

- Qprep will only accept and read ATOM and HETATM records. All other record types have to be removed.
- Residue numbers must be numeric, *i.e.* alphanumeric residue identifiers like 60B are not allowed. Use the renumber script to renumber residues. The numbering does not have to start at 1, but Qprep will renumber residues starting at 1.
- Molecules must be separated with a gap marker line. This line should contain only the word GAP in capital letters, optionally preceded by blanks or tabs. There should not be a gap marker at the end of the file.
- Atom numbering is not significant and Qprep will renumber atoms starting from 1. Note that the numbering will change due to the insertion of hydrogen atoms.
- Only upper case letters may be used in PDB files.
- Residue names must match fragment library entry names. The maximal length of residue names is 3 alphanumeric characters (position 13 to 16). Most other characters like +, -, _ are also permitted.
- Atom names must match atom names within the relevant fragment library entry.
- Temperature factors and occupancy numbers are ignored by Qprep and are optional.

D.3.2 Atom masks

An atom mask defines a subset of atoms and is used:

- in Qprep to select which atoms are included when writing structure files (PDB and mol2)
- in **qdyn** to select which atoms to include in the trajectory and
- in Qcalc to select atoms for superposition and coordinate deviation calculations.

The atom mask is constructed by selecting atoms within a sequence that match a set of properties. The properties that can be selected are:

- solute: all atoms except solvent atoms
- heavy: all atoms except hydrogens (atoms with mass ≥ 4 mass units to be precise)
- excluded: atoms outside the simulation sphere including excluded solvent molecules
- restrained: atoms in the restrained boundary zone and atoms outside the sphere

Each property can be negated by putting 'not' before it. Multiple properties on the same line are combined with a logical and. At the end of a line an atom sequence is defined by its first and last numbers. A sequence of residues can be selected using the word residue before the numbers. The number of the last atom or residue may be omitted to select a single entity. Multiple such lines may be used to construct the mask in an additive manner the atom sets specified by each line are combined with a logical or. If no mask is given, all atoms are selected by default.

D.3.3 Qprep commands

Command	Alias	Arguments (optional)	Description
addbond	ab		Add extra bonds (*e.g.* S-S).
boundary	bc	[boundary condition]	Set the boundary condition (sphere(1), box(2))
changeimp			Redefine (specified) improper torsions.
checkangs	ca	[energy threshold]	Check angle energies.
checkbonds	cb	[energy threshold]	Check bond energies.
checkimps	ci	[energy threshold]	Check improper torsion energies.
checktors	ct	[energy threshold]	Check torsion energies.
clearbond			Clear extra bonds.
clearlib	cl		Clear all loaded molecular libraries.
cleartop			Clear the current topology.
help	h, ?		Show command list.
listres	lr	[residue number]	List atoms in residue.
listseq	ls		List the residue sequence.
makeshell	ms		Fix the mask of the atoms in the restrained shell.
maketop	mt		Generate the topology.
mask	ma	[mask_def] or [none]	Add to or clear atom mask.
preferences	prefs		List atoms in residue.
quit	q		Quit the program.
readframe	rf	[trajectory file [frame no.]]	Load coordinates from Q trajectory file (unformatted)
readlib	rl	[library file]	Read library file. This command may be repeated to load multiple libraries.
readnext	rn		Load next frame of coordinates from current trajectory file
readpdb	rp	[pdb file]	Read pdb file.
readprm	rprm	[parameter file]	Read force field parameter file.
readtop	rt	[topology file]	Read topology file.
readx	rx	[restart file]	Load coordinates from Q restart file (unformatted).

Command	Alias	Arguments (optional)	Description
set	s		Set preferences.
solvate (box)	so	[grid] [solvent name] [boxcentre] [boxsize]	Add solvent molecules from a grid to a box.
		[file name] [boxcentre] [boxsize]	Add solvent from a file with a box of solvent molecules.
		[restart] [solvent name][filename][boxcentre][boxsize]	Add solvent molecules from a restart file containing the same solute with solvent.
solvate (sphere)	so	[centre][radius][grid] [solvent name]	Add solvent molecules from a grid to a sphere.
		[centre][radius][file][file name]	Add solvent from a file with a box or a sphere of solvent molecules.
		[centre][radius][restart] [file name][solvent name]	Add solvent molecules from a restart file containing the same solute with solvent.
trajectory	tr	[trajectory file]	Open trajectory file.
writemol2	wm	[mol. no. [mol2 file [hydrogen flag [water flag]]]]	Write SYBYL mol2 file for one or all molecules in topology using current coordinates. [mol. no.] is the number of the molecule in the topology (separated by a GAP). Enter 0 for all molecules.[mol2 file] is the name of the file to be created. An existing file will be overwritten. Enter auto to generate the name automatically using the template coord_file.frame_no.molecule_no.mol2 [hydrogen flag] specifies whether hydrogens should be written. Enter y for yes or n for no.[water flag] specifies whether water should be written. Enter y for yes or n for no.
writepdb	wp	[pdb file [gap flag]]	Write PDB file containing the atoms specified by the current mask (default all atoms).[gap flag] specifies whether GAP lines between molecules should be written. Enter y for yes or n for no.
writetop	wt	[topology file ["title"]]	Write topology file.

COMMAND	ALIAS	ARGUMENTS (OPTIONAL)	DESCRIPTION
xlink	xl		Search for possible cross-links such as disulphides and make bonds. For each non-bonded heavy atom pair separated by less than 2.1 Å, the program will ask whether to add a bond or not.

D.3.4 Qprep preferences

Use together with set, *e.g.*: set solvent_pack 2.6.

NAME	MEANING	DEFAULT VALUE
solvent_pack	minimum distance between solute and solvent heavy atoms when adding solvent	2.4 Å
solute_density	number density of solute	0.05794 $Å^{-3}$
max_xlink	upper limit of bond length when searching for possible cross-link bonds	2.1 Å
random_seed_solute	integer seed value for the random number sequence used to generate solute hydrogen atom coordinates	179857
random_seed_solvent	integer seed value for the random number sequence used to generate solvent hydrogen atom coordinates	758971

D.3.5 Fragment library file format

The fragment library file is a text file containing definitions of all the molecular building blocks, *i.e.* amino acid residues, ligands etc., and is used by Qprep to generate a molecular topology from a PDB file. The format of the library file follows the same standard as the parameter file. Each fragment/residue entry starts with an entry name enclosed in curly braces, *e.g.* {ALA}, maximum 3 positions. The lists of atoms, bonds, etc. appear as sections within the entry.

The optional [**info**] section contains the keyword SYBYLtype which identifies the SYBYL substructure type (residue or group) for the entry and is used only for writing mol2 files with Qprep. The [**atoms**] section defines the sequential number, name, type and charge of the atoms in the entry. The atom name must match the name used in PDB files to be read, but the order of atoms is not important. In the following sections atoms are identified by their names. The [**bonds**] section lists all bonds within the entry. The optional [**connections**] contains the keywords head and tail which identify the atoms involved in inter-residue bonds (head is bonded to the tail of the preceding residue and tail is bonded to the head of the next residue). Charge groups are defined, one per line, in the [**charge_groups**] section

as lists of atoms starting with the atom designated as switching atom. The tables below describes the format of these sections, and an example file is included as fig. 8.

[**info**]: General information about the fragment.

KEYWORD	VALUE	COMMENT
SYBYLtype	SYBYL substructure type: RESIDUE or GROUP	Optional, default none. Only used for writing Sybyl mol2 files.
PDBtype	PDB substructure type:ATOM or HETATM	Optional, default ATOM. CONECT records in PDB file will be generated for HETATM groups.
Solvent	on or off	If on, this entry is recognised as a solvent.
Density	number density ($Å^{-3}$)	Only used for solvents, to solvate using a grid and to calculate the effective solvent radius.

[**atoms**]: Define atom names, types and partial charges.

COL.	DESCRIPTION
1	sequence number (from 1 to number of atoms)
2	atom name (4 character string)
3	atom type
4	partial charge (e)

[**bonds**]: Define bonds within the fragment.

COL.	DESCRIPTION
1	name of atom 1
2	name of atom 2

[**connections**]: Define sites of inter-residue bonds.

KEYWORD	VALUE	COMMENT
head	name of head atom to which the preceding residue in the sequence is bonded.	Optional, default none. Should not be defined for entries which are entire molecules.
tail	name of tail atom to which the next residue is bonded.	Optional, default none. Should not be defined for entire molecules.

[**build_rules**]: Define rules for generating hydrogen atom coordinates.

COL.	DESCRIPTION
1	type of rule. The only type available is 'torsion'.
2	name of the hydrogen atom to be generated
3	names of atom 2 in the torsion (the atom to which the hydrogen is bonded)
4	name of atom 3 of the torsion
5	name of atom 4 of the torsion
6	target value of the torsion angle formed by the atoms named in columns 2-3-4-5.

[**impropers**]: Define improper torsion angles. This is only used for force fields where impropers are explicitly defined rather than automatically generated (see parameter file).

COL.	DESCRIPTION
1	name of atom 1. Use a + or - before to refer to atoms in previous or next residue.
2	name of atom 2, the central atom to which 1, 3 and 4 are connected
3	name of atom 3 (use +/- as for atom 1)
4	name of atom 4 (use +/- as for atom 1)

[**charge_groups**]: Define charge groups.

COL.	DESCRIPTION
1	name of switching atom
2...	names of other atoms

```
{ALA}                ! Alanine
    [info]
        SYBYLtype  RESIDUE     !SYBYL substructure type
    [atoms]
         1 N      NH1         -0.470 !At. no., name, type, charge
         2 HN     H            0.310 !At. no., name, type, charge
         3 CA     CT1          0.070 !At. no., name, type, charge
         4 HA     HB           0.090 !At. no., name, type, charge
         5 CB     CT3         -0.270 !At. no., name, type, charge
         6 HB1    HA           0.090 !At. no., name, type, charge
         7 HB2    HA           0.090 !At. no., name, type, charge
         8 HB3    HA           0.090 !At. no., name, type, charge
         9 C      C            0.510 !At. no., name, type, charge
        10 O      O           -0.510 !At. no., name, type, charge
    [bonds]
       CB   CA
       N    HN
       N    CA
       O    C
       C    CA
       CA   HA
       CB   HB1
       CB   HB2
       CB   HB3
    [connections]   !how to bond to previous and next
        head     N
        tail     C
    [impropers]
         N     -C     CA     HN
         C     CA     +N      O
    [charge_groups] !charge groups, with switch atom first
        N    HN   CA   HA
        CB   HB1  HB2  HB3
        C    O
```

Figure 8: The CHARMM library entry for alanine

D.3.6 Force field parameter file format

The force field parameter file is a text file, based on the same standard as the FEP file described in section D.6.2 on page 58. It is divided into sections which can appear in any order and which start with a section title enclosed in square brackets. The data in the file is the constants, which are defined for each multiplet of atom types, in

$$V_{pot} = \sum_{bonds} \frac{1}{2} k_b \cdot (r - r_0)^2 + \sum_{angles} \frac{1}{2} k_\theta \left(\theta - \theta_0\right)^2 + \sum_{dihedrals} K_\varphi \cdot (1 + \cos\left(n \cdot \varphi - \delta\right))$$
$$+ \sum_{\substack{improper \\ dihedrals}} \frac{1}{2} k_\xi \left(\xi - \xi_0\right)^2 + \sum_{\substack{atom \\ pairs\, i,j}} \frac{1}{4 \cdot \pi \cdot \varepsilon_0} \cdot q_i \cdot q_j \cdot r_{ij}^{-1} + A_{ij} \cdot r_{ij}^{-12} - B_{ij} \cdot r_{ij}^{-6}$$

The [**atom_types**] section defines a name, and lists Lennard-Jones parameters and mass for each atom type. There are three sets of LJ parameters: set 1 for normal non-bonded interactions, set 2 for pairs of polar atom types (listed in a the section LJ_type2_pairs) and type 3 for pairs of atoms in 1–4 position. The B parameter is the same for sets 2 and 3. Atom type names are alphanumeric. The length of the name is limited to 8 characters.

The section [**atom_aliases**] is used facilitate the transition from numeric atom type names used in earlier versions. In this section alias names may be assigned to atom types defined in the atom_types section, *e.g.* to be able to use the old numeric name as an alias for the new descriptive atom type name in library files.

The [**LJ_type2_pairs**] section lists pairs of polar atom types that should interact with LJ parameters from set 2.

The [**bonds**] section lists force constant and equilibrium distance for bonds between two atoms. It contains one line for each (applicable) unique pair of atom types. The pair 1–2 is equivalent with the pair 2–1, so only one of these should be included. The pairs may appear in any order, but for reasons of readability it is convenient to sort the lines by both atom types and always have the lower atom type number first on each line.

The [**angles**] section lists force constant and equilibrium angle for 3-atom angles. contains one line for each (applicable) unique triplet of atom types. The pair 1–2–3 is equivalent with the pair 3–2–1, so only one of these should be included. The pairs may appear in any order, but for reasons of readability it is convenient to sort the lines by the middle atom type first, then on the left and finally by the right atom type. It is also preferred to have the left atom type less than or equal to the right one (*i.e.* 1–2–3 rather than 3–2–1).

The [**torsions**] section lists parameters for torsion angles. Torsions can be defined by 4 atom types but in many cases only the two middle atoms are significant. The latter case is indicated by "0" (zero) or "?" in column one and four. A torsion defined with four atom types overrides two-atom definitions. The force constant in the cosine-shaped function for the torsion potential is equal to half the barrier height. The periodicity is the number of maxima passed in a full 360 ° rotation. The phase shift divided by the periodicity is the angle where the first maximum should be. The number of paths is the number of ways that a two-atom torsion can be defined, *i.e.* the product of the number of atoms bonded to the two middle atoms. It is used to distribute the force over all the atoms involved. The preferred order of the lines is analogous with the bonds section: sort by the two middle atom types. Multiple torsion potential terms may be defined for the same set of atom types, to enable more complex torsion potentials. All terms are then added together.

The [**impropers**] section lists force constant and equilibrium angle for improper torsion angles, which are modelled by a harmonic potential. The impropers may be defined by two atom types, but in many cases the second type is not used and set to "0" or "?" as in the torsions section.

The [**options**] section contains three keywords. Vdw_rule selects the rule for combining LJ parameters from two atom types and takes the values "geometric" or "arithmetic". Scale_14 is the scaling factor for electrostatic interactions between atoms in 1–4 positions. Switch_atoms selects the cut-off logic for non-bonded interaction: On = use designated switch (central) atoms of charge groups. Off = include the charge groups if any pair of

atoms is within the cut-off distance.

Table 9 lists the data and units for each column in the different sections, and an example is included as a file example on page 50.

Table 9: Parameter file format.

[**atom_types**]: Define atom types

col.	**description**
1	atom type name, max 8 characters
2	Lennard-Jones A parameter for type 1 pairs ($kcal^{1/2} \cdot mol^{-1/2} \cdot Å^6$) for geometric combination or R* ($kcal \cdot mol^{-1} \cdot Å^{12}$) for arithmetic combination)
3	LJ A parameter type 2 ($kcal^{1/2} \cdot mol^{-1/2} \cdot Å^6$) or R* ($kcal \cdot mol^{-1} \cdot Å^{12}$)
4	LJ B parameter type 1 ($kcal^{1/2} \cdot mol^{-1/2} \cdot Å^3$) or ε ($kcal \cdot mol^{-1} \cdot Å^6$)
5	LJ A parameter type 3 ($kcal^{1/2} \cdot mol^{-1/2} \cdot Å^6$) or R* ($kcal \cdot mol^{-1} \cdot Å^{12}$)
6	LJ B parameter type 2 and 3 ($kcal^{1/2} \cdot mol^{-1/2} \cdot Å^3$) or ε($kcal \cdot mol^{-1} \cdot Å^6$)
7	mass of (extended) atom (u)
8	SYBYL atom type (5 character string), optional

[**atom_aliases**]: Define alias names for atom types col. description

col.	**description**
1	alias name, max 8 characters
2	atom type name defined in atom_types section

[**LJ_type2_pairs**]: list pairs of atom types that use the alternate set of LJ parameters col. description

col.	**description**
1	atom type 1
2	atom type 2

[**bonds**]: Define harmonic bond parameters for pairs of atom types col. description

col.	**description**
1	atom type 1
2	atom type 2
3	force constant ($kcal \cdot mol^{-1} \cdot Å^{-2}$)
4	equilibrium length (Å)
5	SYBYL bond type (2 character string), optional

[**angles**]: Define harmonic angle parameters for triplets of atom types col. description.

col.	**description**
1	atom type 1
2	atom type 2
3	atom type 3
4	force constant ($kcal \cdot mol^{-1} \cdot rad^{-2}$)
5	equilibrium angle (°)

[**torsions**]: Define torsion angle parameters for quadruplets or pairs of atom types col. description

col.	description
1	atom type 1 or 0 or ? to match any atom type
2	atom type 2
3	atom type 3
4	atom type 4 or 0 or ? to match any atom type
5	force constant = $1/2 \cdot$ barrier height (kcal $\cdot$ mol^{-1})
6	periodicity (number of maxima per turn)Add a minus sign before to indicate that more components follow on subsequent lines, i. e. for a torsion potential with multiple components all but the last component should be entered with negative periodicity.
7	phase shift (δ/n define the location of first maximum) (°)
8	number of paths

[**impropers**]: Define harmonic improper torsion parameters for pairs of atom types or for single atom types col. description

col.	description
1	atom type 1
2	atom type 2 or 0 or ? to match any atom type
3	force constant (kcal $\cdot$ mol^{-1} $\cdot$ rad^{-2})
4	equilibrium angle (°)

```
* Q-FF params: GROMOS87 parameters
*-------------------------------------------------------
[options] force-field options
vdw_rule       geometric   ! vdW combination rule
scale_14       1.0 ! electrostatic 1-4 scaling factor

[atom_types] atom type definitions
*tac--- ---Avdw1---Avdw2---Bvdw1----Avdw3---Bvdw2&3---mass----SYBYL-name-old-
comment
H.np         0.00      0.00     0.00      0.00     0.00      3.000  H   ! HC   -
non-polar H
C.3        898.00      0.00    23.65    898.00    23.65     12.001  C.3 ! Csp3 -
bare sp3 C

[atom_aliases]
10     H.np
20     C.3

[LJ_type2_pairs] type-2 van der Waals interaction atom type pairs
*-------------------------------------------------------------------------
N.pep   O.SPC
N.pep   Cl-

[bonds] bond types definitions
*iaci iacj   force.c.  dist.     SYBYL
*-------------------------------------------------------------------------
H.np    C.3         700.000   1.090      1
H.np    C.2         700.000   1.090      1

[angles] angle type definitions
* iaci iacj iack  forceK angle0
*-------------------------------------------------------------------
H.np    C.3     H.np       90.00  106.60
H.np    C.3     C.ar6H     90.00  109.50

[torsions] torsion type definitions
*iaci iacj iack iacl  forceK  #minima phase   #paths
*-------------------------------------------------------------------
?       C.3     C.3H    ?          1.400   3.000    0.000    6
O.3s    C.3sH   C.3sH   O.3s       1.400  -3.000    0.000    1

[impropers] improper torsion type definitions
*iaci iacj forceK   imp0
*-------------------------------------------------------------------
C.2     ?         40.000  180.000
```

Figure 9: Example of a parameter file.

D.3.7 Solvent file format

Qprep can solvate a molecular systems by filling empty space in the simulation sphere or box by water molecules taken from a water file. This file is a special PDB-like file containing a box or a sphere of water molecules. The residue name in the water file is used to designate the library entry to use for generating bonds etc.

When using a box for solvation, the box can be replicated in all direction so that a small box can be used to solvate a big simulation sphere or box. Spheres cannot be replicated and must be larger than the intended simulation system. The box or sphere will be translated to the water generation centre, so the origin used in the file is arbitrary. A sphere can not be used to solvate a system intended for simulation with periodic boundaries. The file format is described in Table 10.

Table 10: Water file format

line	**content**
1	For a box: side of the box (Å).For a sphere: radius of the sphere followed by the word **sphere**.
$3 \cdot$ n-1	Coordinates of the oxygen atom of water molecule n in PDB format.
$3 \cdot$ n	Coordinates of the first hydrogen atom of water molecule n in PDB format.
$3 \cdot$ n+1	Coordinates of the second hydrogen atom of water molecule n in PDB format.

D.4 Boundary conditions

D.4.1 Solute boundary restraints

Solute atoms outside the simulation sphere are excluded from non-bonded interactions and are tightly restrained to their initial coordinates by a harmonic potential with a force constant of 200 kcal·mol^{-1}·Å^{-2}. All bonded interactions are evaluated as for atoms inside the sphere. Solute atoms in the outermost shell of the simulation sphere are also restrained to their initial coordinates with a harmonic potential to avoid distortion of bonds across the sphere boundary. The radius of this shell and the force constant is given in the **qdyn** input file (section [**sphere**], keywords shell_radius and shell_force). The restrained shell radius is by default equal to the outer, *i.e.* no atoms will be restrained unless the restrained shell radius is redefined. The force constant has a default value of 10 kcal·mol^{-1}·Å^{-2}.

The grouping of atoms in the inside shell or excluded regions is done before the simulation starts using the initial coordinates from the topology and is not updated during simulation. The initial coordinates for grouping the shell-atoms can also be taken from or from a separate restraint coordinate file (section [**files**], keyword restraint).

D.4.2 Solvent boundary restraints

Solvent molecules near the sphere boundary must be restrained to avoid "evaporation" and to keep the density correct and uniform in the whole sphere. A central value in the radial restraining of water is the effective solvent radius r_w, which is only almost equal to

the simulation sphere radius. It is the solution to the equation $V_{sphere}(r_w) = \frac{4\pi}{3} \cdot r_w^3 = N_p(r_w) \cdot v_p + N_w \cdot v_w$ where $N_p(r_w)$ is the number of heavy solute atoms within a radius r_w from the water centre, v_p is the average volume per heavy atom in proteins (17.3 Å^3), N_w is the (fixed) number of solvent molecules and v_w is the volume of a solvent molecule (29.9 Å^3 for water).

The radial restraining potential has a half-harmonic term that pushes solvent molecules back into the sphere and a Morse-like term that pulls molecules from inside out towards the boundary [29]:

$$V_{solvent}(r) = \begin{cases} \frac{1}{2} \cdot K \cdot (r - r_0)^2 - D_e & \text{if } r > r_0 \\ D_e \cdot \left(\left(e^{(\alpha \cdot (r - r_0))} \right)^2 - 2 \cdot e^{(\alpha \cdot (r - r_0))} \right) & \text{otherwise} \end{cases} \tag{11}$$

where r is the distance from the water centre, K is the force constant of the half-harmonic potential, D_e is the depth ("dissociation energy") of the Morse potential, α the exponential coefficient of the Morse term. r_0 is the effective solvent radius minus the average deviation distance from the minimum of the half-harmonic potential at the current temperature T: $r_0 = r_w - \sqrt{\frac{k_b \cdot T}{K}}$ where r_w is the target solvent radius. The appropriate values of D_e and α depend on r_w and are calculated using empirical functions calibrated to give correct values for water spheres from 12 to 30 Å:

$$\begin{aligned} D_e(r_w) &= 0.26 \frac{kcal}{mol} \cdot e^{\left(-0.19 \frac{1}{\text{Å}} \cdot (r_w - 15\text{Å})\right)} + 0.74 \frac{kcal}{mol} \\ \alpha(r_w) &= 0.20 \frac{1}{\text{Å}} \Big/ \left(1 + e^{\left(0.4 \frac{1}{\text{Å}} \cdot (r_w - 25\text{Å})\right)}\right) + 0.30 \frac{1}{\text{Å}} \end{aligned} \tag{12}$$

K, D_e and α can be set in the input file to override the calculated values used by default (section [**solvent**], keywords radial_force and morse_depth, respectively).

Water molecules (in the topology) that are initially outside the simulation sphere are excluded from the simulation (with respect to non-bonded interactions and restraints).

Polar solvent molecules near boundary will not be randomly oriented like in bulk solvent and a restraining force is required to make the surface solvent molecules follow the probability distribution of angles between radial axis and dipole vector found in the bulk solvent. When a net charge in the Q-atoms polarises the solvent, the distribution of solvent molecule dipole angles changes. This correction of the average polarisation given by Born's formula is taken into account unless disabled by setting charge_correction to off in the input file. The polarisation distribution restraints are applied in three thin shells to minimise the radial dependence of the polarisation which occurs in a single, thicker shell. The outermost shell is 0.5 Å thick, the second 1.0 Å, the third 1.5 Å, so the polarisation is restrained in the outermost 3 Å of the simulation sphere.

The restraining works by sorting the molecules of each shell according to the angle between dipole vector and radial axis and applying a force to each molecule i to adjust the angle towards the angle of molecule i in a sorted sequence of molecules that follow the target distribution. The potential can be written $V_{polarisation}(\Theta_i) = \frac{1}{2} \cdot K_{pol} \cdot \left(\Theta_i - \Theta_i^{target} \right)^2$ where K_{pol} is the force constant. K_{pol} can be set in the input file (keyword polarisation_force in section water), the default value is 20 kcal·mol^{-1}·rad^{-2}.

D.4.3 Periodic boundary conditions

In periodic boundary conditions (PBC) no restraints as described in the previous section is applied to the atoms and no atoms are excluded. This allows larger flexibility in the molecular system but increases runtime. This section contains some things worth to keep in mind when using Q with PBC.

The shape of the box may be cubic or rectangular. The cut-off must never be larger than one half of the shortest side of the box. This also accounts for the constant pressure algorithm. If the cut-off is too big, the program will stop.

Particles are moved across box boundaries in terms of molecules. When simulating a large protein in water solution the option rigid_box_centre should be set to off (false). This means that the box is in each time step centered around the solutes geometrical center. If there is no solute the box will be centred around the geometrical centre of the solvent.

If rigid_box_centre is on, the centre of the box will be the same as given in the topology throughout the simulation. Note that, if solute is present in a simulation like this, each solute molecule must be assigned just one charge group. This is done by changing the library file.

D.4.4 Constant pressure algorithm

The constant pressure algorithm is a combination of molecular dynamics and Monte Carlo volume sampling. A change in volume is chosen randomly $\Delta V = n_{rand} \cdot \Delta V_{max}$ where n_{rand} is a random number between -1 and 1 and ΔV_{max} is the maximum allowed volume displacement in one move. The new volume is defined as $V' = V + \Delta V$, prime indicating the new configuration. The coordinates are then changed, the system is contracted or expanded. The scaling factor for the side length of the box, $l_{x,y,z}$ is $\sqrt[3]{\frac{V'}{V}}$, thus $l'_i = l_i \cdot \sqrt[3]{\frac{V'}{V}}$.

The proportions of the box are maintained, meaning that a rectangular box stays rectangular. The coordinates, $r_{x,y,z}$ of each molecules centre of mass are scaled according to $r'_i = (r_i - ci)\frac{l'_i}{l_i} + c_i$, where c_i is the coordinate of the centre of the box. This variable is included to handle the case when the box centre does not coincide with the origin of the coordinate system. The contraction of expansion is in terms for molecules, not atoms, which means that all intra molecular distances are kept fixed.

After a new configuration has been set, the potential is recalculated. Only the non-bonded interactions need to be taken in to account because the interior of molecules are not changed. The Metropolis sampling equation is $\Delta W = (U'_{pot} - U_{pot}) + P_0(V' - V)$, where P_0 is the target pressure. The new configuration is accepted with probability

$$P(\Delta V) = \begin{cases} e^{-\frac{\Delta W}{k_B T}} & \Delta W > 0 \\ 1 & \Delta W \leq 0 \end{cases} \tag{13}$$

If ΔW is zero or negative the move is always accepted. Otherwise a new random number, $n \in [0, 1]$, is generated and the configuration is accepted if $n \leq e^{-\frac{\Delta W}{k_B T}}$. The acceptance ratio is controlled with the variable max_volume_displacement, which corresponds to ΔV_{max}.

D.5 Units

The units used in Q are the basic units in table 11 and combinations thereof.

Table 11: Units in Q

time	fs
temperature	K
length	Å
energy	kcal·mol^{-1}
charge	e

D.6 File and format descriptions

D.6.1 qdyn input file format

The name of the input file is passed to **qdyn** as the first argument on the command line. The file is divided into sections, each starting with a section heading enclosed in square brackets. Within a section are lines with a keyword and a value, or just values in a defined order. Comments, starting with '!', '#' or '*' may appear after the data on a line or on separate lines anywhere in the file. The order of sections and the order of data within sections is not important (but the order of table 12 is preferred). Many keywords have default values and can be omitted, to make it easier to set up a simple simulation. The use of default values will be shown in the output from **qdyn**. Sections with no required entries are optional.

Table 12: **qdyn** input file format

[**MD**]:Basic simulation data.

keyword	**value**	**comment**
steps	Number of MD steps.	Required.
stepsize	MD stepsize (Δt) (fs).	Required.
temperature	Target temperature (K).	Required.
bath_coupling	Temperature bath relaxation time T (fs)[30].	Optional, default 100 fs. Must be $\geq$ stepsize.
separate_scaling	Enable (on) or disable (off) separate temperature scaling of solute and solvent.	Optional, default on.
random_seed	Integer value to seed the random number generator.	Optional. Used only for random velocities. Change number to get a different set of velocities.
initial_temperature	Temperature (K) of Maxwellian distribution for random velocities.	Optional except when no restart file is used. Use *only* when velocities should be randomised!
shake_solvent	Enable (on) or disable (off) shake constraining of bonds and angles of water.	Optional, default on. We recommend the use of shake for water.
shake_solute	Shake constraining of solute bonds on or off.	Optional, default off.

Table 12: **qdyn** input file format

shake_hydrogens	Shake constraining of bonds to hydrogen atoms in solute and solvent on or off.	Optional, default off.
lrf	LRF Taylor expansion of electrostatic field beyond cut-off radius (on/off).	Optional, default on.

[**PBC**]: Settings for periodic boundary conditions.

rigid_box_centre	Enables the solute to move periodically between boxes.	Optional, default off.
constant_pressure	Enable (on) or disable (off) simulation in the isothermal isobaric ensemble. Set trial interval in the [intervals] section.	Optional, default off.
max_volume_displ	Maximum change in volume in one Monte Carlo step when using the isothermal isobaric ensemble.	Required when constant_pressure is on.
pressure_seed	Seed for the random number generator used to generate new volume conformations in the isothermal isobaric ensemble.	Optional. Use if simulation in isothermal isobaric ensemble is split into several separate input files. Assures good sampling.
pressure	Target pressure when using the isothermal isobaric ensemble.	Optional, default = 1.0 bar. Use only when constant_pressure is on.
put_solute_back_in_box	Enable (on) or disable (off) the putting back of solute molecules in the box. Does not affect energies.	Optional, default on. Disable if measuring diffusion coefficients etc.
put_solvent_back_in_box	Enable (on) or disable (off) the putting back of solvent molecules in the box. Does not affect energies.	Optional, default on. Disable if measuring diffusion coefficients etc.

[**cut-offs**]: Cut-off radii for non-bonded interactions.

solute_solute	Cut-off radius (Å) for solute-solute atom pairs.	Optional, default 10 Å.
solvent_solvent	Cut-off radius (Å) for solvent-solvent.	Optional, default 10 Å.
solute_solvent	Cut-off radius (Å) for solute-solvent.	Optional, default 10 Å.
q_atom	Cut-off radius (Å) for Q-atoms interaction with all atoms.	Optional, default 99 Å, *i.e.* no cut-off. If PBC used, compare to boxlength.
lrf	Cut-off radius (Å) for LRF expansion.	Optional, default 99 Å, *i.e.* no cut-off.

[**sphere**]: Not used in PBC simulations.

shell_force	Force constant (kcal·mol^{-1}·Å^{-2}) for shell restraints.	Optional, default 10 kcal·mol^{-1}·Å^{-2}.
shell_radius	Inner radius of restrained shell (Å).	Optional, default outer shell radius.

Table 12: **qdyn** input file format

exclude_bonded	Flag controlling whether bonded interactions between excluded atoms should be eliminated (on) or retained (off) to reproduce energies in earlier versions.	Optional, default off.

[**solvent**]: Boundary conditions of solvent sphere. Optional, can be omitted in vacuum simulations and must be omitted in PBC simulations.

radius	Target solvent radius.	Optional. Use this only to override the calculated target radius from the topology.
radial_force	Force constant (kcal·mol^{-1}·Å^{-2}) for half-harmonic radial restraint at boundary.	Optional, default 60 kcal·mol^{-1}·Å^{-2}.
polarisation	Polarisation restraints in outer solvent shell on or off.	Optional, default on.
charge_correction	Enable (on) or disable (off) correction of solvent polarisation restraints for total charge of Q-atoms by Born's formula.	Optional, default on.
polarisation_force	Force constant (kcal·mol^{-1}·rad^{-2}) for solvent polarisation restraints.	Optional, default = 20 kcal·mol^{-1}·rad^{-2}.
morse_depth	Depth (dissociation energy, kcal·mol^{-1}) of Morse-type boundary attraction potential.	Optional, default is a function of water radius.

[**intervals**]: Intervals between saving data and updating non-bond lists.

non_bond	Non-bond list (and LRF data) update interval.	Optional, default 10.
output	Interval for printing energy summaries.	Optional, default 10.
temperature	Interval for printing temperature. It will always be printed if it changed by $> 2\%$ since last printed.	Optional, default 10.
energy	Interval for writing Q-atom energies to energy file.	Optional, default 0 (disabled).
trajectory	Interval for writing coordinates to trajectory file.	Optional, default 0 (disabled).
volume_change	Interval for Monte-Carlo trial of new volume.	Optional, default 0 (disable).

[**files**]: File names for input and output.

topology	Topology file name.	Required.
restart	Name of restart file from which initial coordinates and velocities are loaded.	Optional. If absent, initial coordinates are taken from the topology and random velocities are generated (see section MD).
final	Restart file to which the final coordinates and velocities are written.	Required.

Table 12: **qdyn** input file format

trajectory	Trajectory file name.	Optional, except when trajectory interval > 0.
energy	Energy file name.	Optional, except when energy interval > 0.
fep	FEP file name.	Optional, except when lambda values are given.
restraint	Restart file with coordinates to be used for restraining.	Optional. When used, coordinates in the topology will be replaced. This only changes the co-ordinate set used, the restraints must still be specified in *e.g.* the section sequence_restraints.

[**trajectory_atoms**]: which atoms to include in the trajectory.

The data in this section is an atom mask specification, it follows the rules in section Atom masks. Multiple lines may be used.

The following sections do not contain keywords, but data in columns.

[**lambdas**]: λ weights for the FEP states.

column	**description**
1...	Weight for state 1, state 2, ... All λ_i ϵ[0,1] and $\Sigma\lambda_i = 1$.

[**sequence_restraints**]: Restrain sequences of atoms.

1	Number of first atom in sequence.
2	Number of last atom in sequence.
3	Force constant (kcal·mol^{-1}·Å^{-2}) for harmonic potential.
4	Flag for restraining also hydrogens (0=no, 1=yes).
5	Flag for restraining the sequence of atoms to its mass center (2), geometrical centre by the force proportianl to the corresponding atom mass normalized with C12 mass and acting on all atoms (1), or each atom to its initial coordinates (0 or missing).

[**atom_restraints**]: Restrain individual atom positions.

1	Atom number.
2,3,4	Target x y and z coordinates (Å).
5,6,7	Force constants (kcal·mol^{-1}·Å^{-2}) in x y and z directions. With separate force constants for x,y,z this can be used to restrain atoms to lines and planes as well.
8	FEP state where the restraint is active (energies and forces will be scaled by lambda) or 0 = active in all states.

[**distance_restraints**]: Restrain atom-atom distances.

1	Number of first atom.
2	Number of last atom.
3	Lower distance limit for unrestrained region (Å).
4	Upper distance limit for unrestrained region (Å). Set lower limit = upper limit for standard harmonic potential.
5	Force constant (kcal·mol^{-1}·Å^{-2}).

Table 12: **qdyn** input file format

6	FEP state where the restraint is active (energies and forces will be scaled by lambda) or 0 = active in all states.

[**wall_restraints**]: Elastic wall (half-harmonic) restraints of sequences of atoms.

1	Number of first atom in sequence.
2	Number of last atom in sequence.
3	Distance from water centre beyond which the restraining potential is applied.
4	Force constant (kcal·mol^{-1}·Å^{-2}) for harmonic potential.
5	Constant D_e in the Morse potential, depth of the potential energy minimum.
6	Constant a in the Morse potential.
7	Flag for restraining also hydrogens (0=no, 1=yes).

D.6.2 FEP file format

The FEP file which designates some atoms from the topology as Q-atoms and redefines the topology for these atoms in a number of states between which transformations can be made, is a text file. It is divided into sections which can appear in any order and which start with a section title. Within each section the data appears either as lines with values or as pairs of keyword and value.

Section titles are enclosed in square brackets and must be the first (non-white space) item on a line. They are not case-sensitive. Keyword-value pairs appear in that order, anywhere on a line but together on the same line. Keyword-value lines can appear in any order (within a section). White space is not significant in value-list lines.

Comments start with "!", "#" or "*" and may appear after values or as separate lines.

Only the section atoms where Q-atoms are designated, is required, all others are optional. They may appear in any order, but the preferred order is that of table 13. The section [**FEP**] contains the keyword states followed by the number of FEP states defined in the FEP file. An offset value to add to all topology atom numbers in order to avoid renumbering all atoms between *e.g.* free ligand and bound ligand simulations can also be defined in the [**FEP**] section. New atom types for Q-atoms are defined in the [**atom_types**] section. The assignment of Q-atom types to Q-atoms is done, for each state, in the [**change_atoms**] section. Pairs of atoms between which bonds are made or broken should use the exponential repulsion non-bonded potential instead of the standard Lennard-Jones by listing them under the [**soft_pairs**] heading. In some cases it is desirable to completely turn off certain non-bonded interactions. This can be done on a per-state basis in the section [**excluded_pairs**].

New bond, angle, torsion and improper types can also be defined and used for any atoms in the topology, not only between Q-atoms. Atoms are therefore referred to by their number in the topology rather than by a Q-atom number. Definitions in the topology are overridden by definitions in the FEP file. To disable an interaction in one state, a zero should be used in place of the type number for that state.

Angles, torsions and impropers which depend on the existence of a bond being formed or broken should be "coupled" to that bond by scaling the angle energy by the ratio of the actual value of the Morse bond energy to the dissociation energy. These couplings are defined in the sections [**angle_couplings**], [**torsion_couplings**] and [**improper_couplings**].

Extra shake constraints can be imposed between any pair of atoms in the topology using the heading [**shake_constraints**]. The effective constraint distance will be the sum of the distances for each state weighted by their respective λ's.

Quantum-mechanical mixing of states used in EVB calculations by introducing off-diagonal Hamiltonian matrix element functions is defined in the section [**off_diagonals**].

If vanishing or appearing atoms are part of your FEP strategy, it may be desirable to use a *softcore* potential for the Q-atoms in question.[31] Softcore potentials have been implemented in **qdyn** according to equation 14. Depending on the value of **softcore_use_max_potential** given in the [**FEP**] section, the α-values are either read directly from the [**softcore**] section or calculated by **qdyn** in a pairwise manner based upon the desired potentials at $r = 0$ given in the [**softcore**] section. Softcore potentials are only available for Q-atoms, i.e. Q-Q, Q-water and Q-solute interactions are treated with softcore. In the case of a Q-Q interaction where both Q-atoms have softcore potentials, the α which is used is that which gives rise to the lowest potential at $r = 0$.

$$V_{vdW}(r_{ij}) = \frac{A_{ij}}{(r_{ij}^6 + \alpha)^2} - \frac{B_{ij}}{r_{ij}^6 + \alpha} \qquad \text{or} \qquad V_{vdW}(r_{ij}) = \epsilon \cdot \left(\frac{R_{ij}^{*\,12}}{(r_{ij}^6 + \alpha)^2} - 2 \cdot \frac{R_{ij}^{*\,6}}{r_{ij}^6 + \alpha}\right) \quad (14)$$

If periodic boundary conditions are used an additional section [**PBC**] is needed. In this section one switching atom for all Q-atoms is defined. This switching atom is used when generating the Q-surrounding nonbonded pair lists.

Table 13 lists the data and units for each column in the different sections, and an example is included as file example on page 22.

Table 13: FEP file format

[**atoms**]: Define Q-atoms.

column	**description**
1	Q-atom number (counting from 1 up).
2	Topology atom number.

[**PBC**]: For periodic boundary conditions.

keyword	**value**	**comment**
switching-_atom	Topology atom number.	Required with periodic boundary conditions. Defines which atom to use as switching atom when calculating nonbonded pairlist for qatom interactions

[**FEP**]: General perturbation information.

keyword	**value**	**comment**
states	Number of FEP/EVB states.	Optional, default 1.

Table 13: FEP file format

offset	Topology atom number.	Optional, default 0. This number is added to all topology atom numbers given in the FEP file.
offset_residue	Residue/fragment number.	Optional. Set offset to the topology number of the first atom in the given residue minus one.
offset_name	Residue/fragment name.	Optional. Set offset to the topology number of the first atom in the first residue with the given name minus one.
qq_use_library_charges	This is a special feature for studying *e.g.* electrostatic linear response. Set to 'on' to use the library charges from the topology for intra-Q-atom interactions, i. e. change only Q-atom-surrounding electrostatic interactions.	Optional, default off.
softcore_use_max_potential	Set to 'on' if the values entered in the [**softcore**] section are the desired maximum potentials (kcal/mol) at $r = 0$. **qdyn** will then calculate pairwise α_{ij} to be used in equation 14. 'off' means the values are to be used directly in equation 14.	Optional, default off.

[**change_charges**]: Redefine charges of Q-atoms.

column	**description**
1	Q-atom number (referring to numbering in atoms section).
2...	Charge (e) in state 1, state 2, ...

[**atom_types**]: Define new atom types for Q-atoms: Standard LJ parameters and parameters for the exponential repulsion potential $V_{soft} = C_i \cdot C_j \epsilon^{-a_i \cdot a_j \cdot r_{i,j})}$.

1	Name (max 8 characters).
2	Lennard-Jones A parameter ($\text{kcal}^{\frac{1}{2}} \cdot \text{mol}^{-\frac{1}{2}} \cdot \text{Å}^6$) for geometric combination or R* ($\text{kcal} \cdot \text{mol}^{-1} \cdot \text{Å}^{12}$)for arithmetic combination rule.
3	LJ B parameter ($\text{kcal}^{\frac{1}{2}} \cdot \text{mol}^{-\frac{1}{2}} \cdot \text{Å}^3$) or ϵ ($\text{kcal} \cdot \text{mol}^{-1} \cdot \text{Å}^6$).
4	Soft repulsion force constant C_i ($\text{kcal}^{\frac{1}{2}} \cdot \text{mol}^{-\frac{1}{2}}$) in V_{soft}.
5	Soft repulsion distance dependence parameter a_i ($\text{Å}^{-\frac{1}{2}}$) in V_{soft}.
6	Lennard-Jones A parameter ($\text{kcal}^{\frac{1}{2}} \cdot \text{mol}^{-\frac{1}{2}} \cdot \text{Å}^6$) or R* ($\text{kcal} \cdot \text{mol}^{-1} \cdot \text{Å}^{12}$) for 1-4 interactions.
7	LJ B parameter ($\text{kcal}^{\frac{1}{2}} \cdot \text{mol}^{-\frac{1}{2}} \cdot \text{Å}^3$) or e ($\text{kcal} \cdot \text{mol}^{-1} \cdot \text{Å}^6$) for 1-4 interactions.
8	Atomic mass (u).

[**change_atoms**]: Assign Q-atom types to Q-atoms.

1	Q-atom number.
2...	Q-atom type name in state 1, state 2, ...

[**soft_pairs**]: Define pairs which use soft repulsion.

1	Q-atom number of first atom in pair.
2	Q-atom number of second atom in pair.

Table 13: FEP file format

[**excluded_pairs**]: Define pairs to exclude from non-bonded interactions. Note: also non-Q-atoms can be excluded.

1	Topology atom number of first atom in pair.
2	Topology atom number of second atom in pair.
3...	Exclusion effective (1) or not (0) in state 1, state 2, ...

[**el_scale**]: Define q-atom pairs for scaling of the electrostatic interaction. Can be useful e.g. when highly charged intermediates appear in FEP/EVB. The scale factor applies to all states. Note: only Q-atom pairs can be scaled.

1	q-atom number of first atom in pair
2	q-atom number of second atom in pair
3	electrostatic scale factor (0..1)

[**softcore**]: Define q-atom softcore potentials. The meaning of these entries depends on the value of softcore_use_max_potential.

1	q-atom number
2...	Desired potential at $r = 0$ for all of this q-atom's vdW interactions in state 1, state 2, ... or the actual α value used in equation 14. An α of 200 yields vdW potentials at $r = 0$ of 10-50 kcal/mol for heavy atom - heavy atom interactions. Set to 0 if softcore is not desired for this q-atom.

[**monitor_groups**]: Define atom groups whose non-bonded interactions are to be monitored (printed in the log file).

1...	Topology atom number of first and following atoms in group.

[**monitor_group_pairs**]: Define pairs of monitor_groups whose total non-bonded interactions should be calculated.

1	First monitor_group number.
2	Second monitor_group number.

[**bond_types**]: Define Q-bond types using Morse or harmonic potentials,
$E_{Morse} = D_e \left(1 - e^{-\alpha(r-r_0)}\right)^2$ $E_{Harmonic} = \frac{1}{2} k_b \left(r - r_0\right)^2$.
Morse and harmonic potentials can be mixed (but each bond type is either kind). Entries with four values are Morse potentials and entries with three values are harmonic.

	Morse potential	**Harmonic potential**
1	Q-bond type number (starting with 1).	
2	Morse potential dissociation energy, D_e (kcal·mol^{-1}).	Harmonic force constant k_b (kcal·mol^{-1}·Å^{-2}).
3	Exponential co-efficient α in Morse potential (Å^{-2}).	Equilibrium bond length r_0 in harmonic potential (Å).
4	Equilibrium bond length r_0 in Morse potential (Å).	

[**change_bonds**]: Assign Q-bond types. Note: shake constraints for the redefined bonds are removed. The order in which atoms are given is not important.

1	Topology atom number of first atom in bond.
2	Topology atom number of second atom in bond.
3...	Q-bond type number (referring to numbering in bond_types section) or 0 to disable bond in state 1, state 2, ...

Table 13: FEP file format

[**angle_types**]: Define Q-angle types.

1	Q-angle type number (starting with 1).
2	Harmonic force constant (kcal·mol^{-1}·rad^{-2}).
3	Equilibrium angle (°).

[**change_angles**]: Assign Q-angle types.

1	Topology atom number of first atom in angle.
2	Topology atom number of middle atom in angle.
3	Topology atom number of third atom in angle.
4...	Q-angle type number (referring to numbering in angle_types section) or 0 to disable angle in state 1, state 2, ...

[**torsion_types**]: Define Q-torsion types.

1	Q-torsion type number (starting with 1).
2	Force constant = $\frac{1}{2}$·barrier height (kcal·mol^{-1}).
3	Periodicity (number of maxima per turn).
4	Phase shift (°).

[**change_torsions**]: Assign Q-torsion types. Note: The order of atoms (1, 2, 3, 4 or 4, 3, 2, 1) is not important.

1	Topology atom number of first atom in torsion.
2	Topology atom number of second atom in torsion.
3	Topology atom number of third atom in torsion.
4	Topology atom number of fourth atom in torsion.
5...	Q-torsion type number (referring to numbering in torsion_types section) or 0 to disable torsion in state 1, state 2, ...

[**improper_types**]: Define Q-improper types.

1	Q-improper type number (starting with 1).
2	Harmonic force constant (kcal·mol^{-1}·rad^{-2}). N.B. new impropers defined here are always harmonic.
3	Equilibrium angle (°).

[**change_impropers**]: Assign Q-improper types. Note: The order of atoms (1, 2, 3, 4 or 4, 3, 2, 1) is not important.

1	Topology atom number of first atom in improper.
2	Topology atom number of second atom in improper.
3	Topology atom number of third atom in improper.
4	Topology atom number of fourth atom in improper.
5...	Q-improper type number (referring to numbering in improper_types section) or 0 to disable improper in state 1, state 2, ...

[**angle_couplings**]: Couple Q-angles to Q-bonds, *i.e.* scale angle energy by the ratio of the actual value of the Morse bond energy to the dissociation energy.

1	Q-angle number (line number within change_angles section).
2	Q-bond number (line number within change_bonds section).

[**torsion_couplings**]: Couple Q-torsions to Q-bonds.

1	Q-torsion number (line number within change_torsions section).
2	Q-bond number (line number within change_bonds section).

Table 13: FEP file format

[**improper_couplings**]: Couple Q-impropers to Q-bonds.

1	Q-improper number (line number within change_impropers section).
2	Q-bond number (line number within change_bonds section).

[**shake_constraints**]: Define extra shake constraints. The effective constraint distance will be the sum of the distances given for each state, weighted by their λ values. Note: constraints defined here do not override constraints imposed by setting the shake flag to *on* in the **qdyn** input file. To remove a constraint the bond must be redefined as a Q-bond. The order in which atoms are given is not important.

1	Topology atom number of first atom.
2	Topology atom number of second atom.
3...	Constraint distance (Å) in state 1, state 2, ...

[**off-diagonals**]: Define off-diagonal elements of the Hamiltonian, represented by $H_{i,j} = A_{i,j} \cdot \epsilon^{-\mu_{i,j} \cdot r_{k,l}}$ where i and j are states and k and l are Q-atoms.

1	State i.
2	State j.
3	Q-atom k.
4	Q-atom l.
5	$A_{i,j}$ (kcal·mol^{-1}).
6	$\mu_{i,j}$ (Å^{-1}).

D.7 Log files

D.7.1 qdyn log file

The information from the **qdyn** simulation is gathered in a log file. This log file is divided into two parts, one showing the initialisation of the simulations and the other showing different energies form all the steps of the simulation. The units used in the log files are the basic units in table 11 on page 54 and appropriate combinations thereof.

Initialisation phase

In this part of the log file, all input data from input, topology and FEP files are read and the simulation is initialised.

Reading input from eq1.inp

Number of MD steps = 1 Stepsize (fs) = 0.200	
Target temperature = 1.00 T-relax time = 1.00	Temperature bath temperature and time constant for coupling.
Initial velocities will be generated from Maxwell distribution: Maxwell temperature= 1.00 Random number seed= 4320	Initial velocities are random values from Maxwell's velocity dirstibution at 1 K.
Shake constraints on all solvent bonds: on Shake constaints on all solute bonds: off Shake constaints on all bonds to hydrogen: off	
Nonbonded method = LRF Taylor expansion outside cut-off	Local reaction field approximation applied beyond cut-off.
Cut-off radii for non-bonded interactions: Solute-solute: 10.00 Solvent-solvent: 10.00 Solute-solvent: 10.00 Q-atom-non-Q-atom: 99.00 LRF: 99.00	
Shell restraint force constant = 50.00	Heavy solute atoms in the shell region defined in the topology will be restrained to their initial positions by harmonic potentials with force constant 50 kcal·mol^{-1}·Å^{-2}.
Water polarisation force constant set to default	

Non-bonded pair list update interval = 25	
Energy summary print-out interval = 10 Temperature print-out interval = 1 Trajectory write interval = 5 Energy file write interval = 5	Energy summary will be written to log file every 10 steps, Temperature every step, a trajectory frame every 10 steps, and an energy file record every 5 steps.
Topology file = cdc25v4.top	
Initial coordinates taken from topology.	The simulation starts from topology coordinates, not from a restart file.
Final coord. file = eq1.re Trajectory file = eq1.dcd Energy file = eq1.en FEP input file = pt.fep	
lambda-values = 0.50000 0.50000	Weights for the 2 FEP states used.
Listing of restraining data:	
No. of sequence restraints = 1 atom_i atom_j fc H-flag to_centre 1 1619 5.00 0 0	Atoms 1 through 1619 will be restrained to their initial positions with force constant 5 kcal·mol^{-1}·Å^{-2} but hydrogens will not be restrained. Restraints will be applied to individual atom positions, not to the geometrical centre of the set of atoms.
No. of distance restraints = 6 atom_i atom_j dist1 dist2 fc state 1631 1624 1.50 1.50 5.00 0 1631 922 1.50 1.50 5.00 0 922 1624 3.00 4.00 5.00 0 911 924 3.20 3.20 10.00 0 912 924 2.20 2.60 10.00 0 932 1621 3.20 3.20 10.00 0	Six atom-atom distances will be restrained using harmonic potentials (dist1=dist2) or flat-bottom potentials (0 between dist1 and dist2) with different force constants. The zero means that the restraints will be in effect in all states.

Reading topology file

CDC25 + PP1 + 2 H2O	Title of topology.
No. of solute atoms = 1631 No. of solvent atoms = 882	There are 1631 solute atoms.
No. of coordinates = 7539 No. of atom type codes = 2513 No. of solute bonds = 1670 No. of solvent bonds = 882 No. of solute angles = 2380 No. of solvent angles = 294 No. of solute torsions = 3037 No. of solvent torsions = 0 No. of solute impropers = 626 No. of solvent impropes = 0 No. of atomic charges = 2513 No. of solute charge grps = 1036 No. of solvent charge gps = 294	
vdW rule [1=G / 2=A] = 1 El-static 1-4 damping = 1.000 Coulomb constant = 332.0000	The force field uses the geometric combinatin rule for LJ parameters, no recuction of electrostaic 1-4 interactions and the constant in Coulomb's law is 332 kcal·mol^{-1}·e^{-2}·Å.
No. of atom types = 32 No. of LJ type 2 pairs = 87	
No. of heavy atoms = 1342	
No. of 1-4 neighbours = 2962 No. long-range 1-4 nbrs = 0	The 1-4 neighbour list.
No. of nbor exclusions = 4932 No. of long-range excls = 0	The 1-2 & 1-3 neighbour list is also divided into two parts.
No. of residues = 456 No of solute residues = 162	There are 162 solute fragments and 294 water molecules, giving 456 fragments.
No. of molecules = 296	The are 296 separate molecules (protein+ligand+294 waters).
Atom type names = 32 SYBYL atom types = 32	Atom type names and their SYBYL mol2 file equivalents.
Exclusion radius = 16.000 Restrained shell radius = 14.500 Eff. solvent radius = 15.920 Solute centre = 11.040 42.102 67.730 Solvent centre = 11.040 42.102 67.730 No. of excluded atoms = 1051 No. of atoms in shell = 114	Radius of simulation sphere. Inner radius of restrained shell. Effective solvent radius (used for solvent restraints) based on the number and distribution of solute and solvent atoms. Coordinates for the centre of the simulation sphere and the solvent sphere. 1051 atoms are outside the simulation sphere and excluded from bonded interactions. 114 atoms are in the restrained shell.
Molecular topology read successfully.	

Reading Q atom list

No. of fep/evb states = 2 No. of fep/evb atoms = 8	
Atom nos.: 921 922 1631 1620 1621 1622 1623 1624	These atoms in the topology become Q atoms.

Reading fep/evb strategy

No. of changing charges = 8	Charges of 8 Q atoms are to be changed.

Effective Q-atom charges for all Q-atoms

Q atom	charge in state 1	state 2
1	0.180	0.000
2	-0.450	-1.000
3	0.270	0.398
4	0.940	1.230
5	-0.360	-0.360
6	-0.860	-0.860
7	-0.860	-0.860
8	-0.860	-0.548
SUM	-2.000	-2.000

The charges of ALL Q atoms are listed, whether changed or not.

The sum of the Q atom charges is important to check!

No. of Q-atom types = 7

7 new atom types are to be defined.

Name	Ai	Bi	Ci	ai	Ai(1-4)	Bi(1-4)	Mass
P	2303.00	59.35	0.00	1.81	2303.00	59.35	30.97
OE	600.00	23.25	70.00	1.81	600.00	23.25	16.00
OD	956.00	23.01	70.00	1.81	550.00	23.25	16.00
H	0.00	0.00	6.50	1.81	0.00	0.00	3.00
CB	2906.00	46.63	0.00	0.00	1304.00	33.60	14.03
SH	2001.57	44.74	165.00	1.81	2001.57	44.74	32.06
S-	7200.00	136.00	165.00	1.81	2001.57	44.74	32.06

Parameters for Q atom types.

Assigning Q-atom types to all Q atoms:

Q atom	atom type in state 1	state 2
1	CB	CB
2	SH	S-
3	H	H
4	P	P
5	OE	OE
6	OD	OD
7	OD	OD
8	OD	OE

Atom types for ALL Q atoms are redefined in each state.

No. of soft repulsion non-bonded pairs = 2

atom_i	atom_j
2	3
3	8

Two pairs of non-bonded atoms should interact with the exponential repulsion potential instead of Lennard-Jones.

No. of excluded non-bonded pairs = 3

atom_i	atom_j	excluded in state 1	state 2
1631	1621	0	1
1631	1622	0	1
1631	1623	0	1

Three pairs of non-bonded atoms should not interact at all.

Q-bond types:

type #	Morse	E_diss	alpha	b0	Harmonic force_k
1	85.00	2.00	1.61		
2	120.00	2.00	1.49		
3	110.00	2.00	1.00		
4	94.00	2.00	1.33		

Four Morse type bond potentials are defined. The first one has a dissociation energy of 85 kcal/mol, exponential coefficient 2 Å-2 and equilibrium bond length 1.61 Å.

No. of changing bonds = 3

atom_i	atom_j	bond type in state 1	state 2
922	1631	4	0
1624	1631	0	3
1620	1624	2	1

Three bonds are redefined using the Morse bond types above. Bond type zero in one state means the bond is not present in that state.

Q-angle types:

type #	force-k	theta0
1	95.00	109.50
2	140.00	120.00
3	110.00	109.50
4	95.00	96.00

Four angle potentials are defined. The first one has a harmonic force constant of 95 $\text{kcal}\cdot\text{mol}^{-1}\cdot\text{rad}^{-2}$ and equilibrium angle 109.5°.

No. of changing angles = 4

atom_i	atom_j	atom_k	angle type in state 1	state 2
921	922	1631	4	0
1631	1624	1620	0	3
1624	1620	1622	2	1
1624	1620	1623	2	1

Four angles are redefined using the angle types above. Angle type zero in one state means the angle is not present in that state.

Q-torsion types:

type #	force-k	mult	delta
1	0.70	3.00	0.00
2	0.25	-3.00	0.00
3	0.75	2.00	0.00

Three torsion potentials are definedThe first one has a rotation barrier half-height of 0.7 kcal/mol, 3 maxima per turn and the first maximum at 0°.

No. of changing torsions = 7

atom_i	atom_j	atom_k	atom_l	torsion type in state 1	state 2
920	921	922	1631	1	0
1631	1624	1620	1621	0	2
1631	1624	1620	1621	0	3
1631	1624	1620	1622	0	2
1631	1624	1620	1622	0	3
1631	1624	1620	1623	0	2
1631	1624	1620	1623	0	3

Seven torsions are redefined using the torsion types above. torsion code zero in one state means the torsion is not present in that state.

Q-improper types:

type #	force-k	imp0
1	95.00	120.00

One improper torsion potential is defined. It has a harmonic force constant of 95 $\text{kcal}\cdot\text{mol}^{-1}\cdot\text{rad}^{-2}$ and its minimum at 120°.

No. of changing impropers= 1

atom_i	atom_j	atom_k	atom_l	improper type in state 1	state 2
920	921	922	1631	1	0

One improper torsion is redefined using the improper type defined above. Improper code zero in one state means the improper is not present in that state.

No. of angle-Morse couplings = 2

angle_i	bond_j
1	1
2	2

Two Q angles are to be coupled to Q bonds.

No. of torsion-Morse couplings = 7 torsion_i bond_j 1 1 2 2 3 2 4 2 5 2 6 2 7 2	Seven Q torsions are to be coupled to Q bonds.
No. offdiagonal (Hij) funcs. = 1 state_i state_j atom_k atom_l Aij mu_ij 1 2 2 8 1.00 0.45	One off-diagonal Hamiltonian functions for mixing of states is defined. It relates states 1 and 2 by a function of the distance between Q atoms 2 and 8 and the parameters A=1.00 kcal/mol and μ=0.45 Å^{-1}
No. atom groups to monitor = 2 group 1: 921 922 group 2: 1620 1621 1622 1623 1624	Two groups of atoms are defined for monitoring their non-bonded interactions.
No. of group pairs to monitor = 1 group_i group_j 1 2	The non-bonded interactions between atom group 1 and atom group 2 above are to be monitored.

Removing redefined interactions from topology

type atom1 atom2 atom3 atom4 bond 1620 1624 bond 1624 1631 angle 1622 1620 1624 angle 1623 1620 1624 angle 1620 1624 1631 torsion 1621 1620 1624 1631 torsion 1621 1620 1624 1631 torsion 1622 1620 1624 1631 torsion 1622 1620 1624 1631 torsion 1623 1620 1624 1631 torsion 1623 1620 1624 1631	Bonds, angles, torsions and impropers redefined by the FEP file are removed from the normal topology.

Initialising dynamics

Total charge of non-Q atoms = 1.00 Total charge of system = -1.00	The sum of the partial charges of all non-Q atoms is 1.00. Including also the effective charge of the Q atoms gives the total charge -1.00 for the whole system.
Target water sphere radius = 15.92	This radius is calculated from the number of water molecules and solute atoms and the average number densities, to ensure correct density of the water.
Surface inward harmonic force constant = 60.00 Surface attraction well depth = 0.96 Surface attraction well width = 0.49	Force constant in the radial half-harmonic potential acting on water molecules outside the target water sphere radius and parameters for the boundary attraction potential. These are normally calculated by the program, based on the radius.
Water polarisation restraints : ON, Born correction enabled Radial polarisation force constant = 20.00	Water polarisation restraints in the boundary region are enabled and the polarisation will be corrected for the net charge of the system.
Setting up 3 water shells for polarisation restraints. Shell # outer radius inner radius 1 15.92 15.42 2 15.42 14.42 3 14.42 12.92	The polarisation restraints are applied in three sub-shells.
Coordinates for 472 atoms will be written to the trajectory.	The atom mask specification in the trajectory_atoms section of the input file matched 472 atoms.
Number of shake constraints = 2552 No. molecules with shake constraints = 296 Initial x-shaking required 3 interations per molecule on average. Initial v-shaking required 8 interations per molecule on average.	Initial positions and velocities are shaken.

Nonbonded pair count and distribution

node value solute-solute solute-water water-water Q-solute Q-water all count 35222 39304 118980 4576 9648 0 alloc 37183 41469 126720 4904 9648	Non-bonded pairs are counted and pair lists are allocated dynamically (with a little extra space). The parallel version of **qdyn** will list pair list sizes also for other nodes than node 0.
No. of Rcq indep. nb pairs involving q-atoms = 55 in state : 1 No. of Rcq indep. nb pairs involving q-atoms = 52 in state : 2	The size of the Q-atom-Q-atom non-bonded pair list in each state. This list also includes interactions between Q atoms and non-Q atoms which are bonded, angled or torsioned to a Q atom.

Initial temperatures are : Ttot = 1.08 Tfree = 1.14	The initial temperature is calculated. Ttot takes all the degrees of freedom of the system into account, whereas Tfree excludes restrained degrees of freedom, *e.g.* excluded atoms and shake constraints.

Simulation phase

This part of the log file shows the progress of the simulation in terms of temperatures and energy summaries.

Nonbonded pair list generation

node	value	solute-solute	solute-water	water-water	Q-solute	Q-water	Non-bonded pair lists are generated. In the parallel version, each node reports its list sizes.
0 count	35222	39304	118989	4528	9648		

Energy summary at step 0

	el	vdW	bond	angle	torsion	improper	Bond and angle energies are zero due to shake.The Q-atom line lists the effective, *i.e.* λ-weighted Q-atom energies. The total restraint energy is a sum of terms from fixing excluded atoms, the radial water restraints, the water polarisation restraints, restraining of solute atoms in the shell near the boundary and solute restraints specified in the input file.The total energy of the system is the sum of potential and kinetic energy.
solute	-560.13	-182.55	0.00	0.00	309.92	96.67	
solvent	-3402.39	549.41	0.00	0.00	0.00	0.00	
solute-solvent	-494.15	383.01					
LRF	40.70						
Q-atom	-420.84	78.87	-190.09	3.43	1.89	0.00	
	total	fix	water_rad	water_pol	shell	solute	
restraints	46.05	0.00	4.36	19.06	0.00	22.63	
	total	potential	kinetic				
SUM	-3065.30	-3075.23	9.93				

Q-atom energies at step 0

type	st	lambda	el	vdW	bond	angle	torsion	improper	The Q atom energies are calculated and shown for each state, together with the λ values. Q atom-Q atom non-bonded energies also include interactions with non-Q atoms bonded, angled or torsioned to Q atoms. Q-surrounding interaction energies are the sum of Q-solute ("Q-prot") and Q-water energies. These energies enter into LIE calculations of binding affinity. This is the sum of all the above. Restraints that are applied per state, *e.g.* positional and distance, are listed for each state. The total Q-atom energies are the sum of all the above and the restraints.
Q-Q	1	0.5000	-45.93	40.43					
Q-Q	2	0.5000	52.53	51.53					
Q-prot	1	0.5000	-330.52	34.00					
Q-prot	2	0.5000	-315.82	33.62					
Q-wat	1	0.5000	-110.72	-0.75					
Q-wat	2	0.5000	-91.21	-1.09					
Q-surr.	1	0.5000	-441.25	33.25					
Q-surr.	2	0.5000	-407.03	32.53					
Q-any	1	0.5000	-487.17	73.68	-194.18	6.69	0.49	0.00	
Q-any	2	0.5000	-354.50	84.06	-186.00	0.17	3.30	0.00	
type	st	lambda	total	restraint					
Q-SUM	1	0.5000	-577.86	22.63					
Q-SUM	2	0.5000	-430.34	22.63					
H(1, 2) = 0.32 dist. between Q-atoms 2 8 = 2.56									Off-diagonal Hamiltonian function value and the inter-atomic distance on which it is based.

Monitoring selected groups of nonbonded interactions

pair	Vwsum	Vwel	Vwvdw	1:Vel	1:Vvdw	2:Vel	2:Vvdw	The non-bonded interaction energies between the pairs of atom groups specified in the FEP file are listed both as λ-weighted sums and for each state separately.
1	98.16	95.99	2.17	58.64	1.97	133.35	2.37	

Temperatures

Temperature at step 1: T_tot= 1.4 T_free= 1.6	The total and "free" temperatures are printed at the specified interval, or if the total temperature changed by >2% since it was last printed. This simulation was only one step.

FINAL Energy summary

	el	vdW	bond	angle	torsion	improper
solute	-560.13	-182.55	294.99	369.97	309.92	96.67
solvent	-3402.39	549.41	0.00	0.00	0.00	0.00
solute-solvent	-494.15	383.01				
LRF	40.70					
Q-atom	-420.84	78.87	-190.09	3.43	1.89	0.00
	total	fix	water_rad	water_pol	shell	protein
restraints	46.05	0.00	4.36	19.06	0.00	22.63
	total	potential	kinetic			
SUM	-3065.30	-3075.23	9.93			

The energy summaries at the end of the simulations contain the same information as the ones printed during the simulations.

FINAL Q-atom energies

type	st	lambda	el	vdW	bond	angle	torsion	improper
Q-Q	1	0.5000	-45.93	40.43				
Q-Q	2	0.5000	52.53	51.53				
Q-prot	1	0.5000	-330.52	34.00				
Q-prot	2	0.5000	-315.82	33.62				
Q-wat	1	0.5000	-110.72	-0.75				
Q-wat	2	0.5000	-91.21	-1.09				
Q-surr.	1	0.5000	-441.25	33.25				
Q-surr.	2	0.5000	-407.03	32.53				
Q-any	1	0.5000	-487.17	73.68	-194.18	6.69	0.49	0.00
Q-any	2	0.5000	-354.50	84.06	-186.00	0.17	3.30	0.00

type	st	lambda	total	restraint
Q-SUM	1	0.5000	-577.86	22.63
Q-SUM	2	0.5000	-430.34	22.63

H(1, 2) = 0.32 dist. between Q-atoms 2 8 = 2.56

qdyn version 4.15 terminated normally.

This simulation complete without errors.

D.7.2 Qfep log file

[To be added]

D.8 Differences from earlier versions

FIX THIS SECTION! Remove, update or put it on the web!

This section is only of interest for users of Q version 3 and presents the important differences between versions 3 and 4. The trajectory analysis program Qcalc is added to the package, making it easy to calculate RMS coordinate deviations (with or without least squares fitting of structures), distances, angles and individual bonded force field terms.

D.8.1 Differences in Qprep

- Solvation is done in Qprep, not in **qdyn**. The topology thus contains all the atoms of the simulated system.
- Solvent can be generated using a grid, from a solvent coordinate file or from a restart file.
- The simulation sphere is specified when generating the topology and exclusion of atoms takes place in Qprep and is included in the topology.

- Hydrogen atom coordinates are generated by steepest descent energy minimisation of the angle potentials involving the hydrogens, rather than by determining the hybridisation by heuristic rules.
- The placement of hydrogen atoms may be further controlled by build rules where a torsion angle can be specified.
- Shake constraints on water angles now require that a H-H bond is present in the library entry. Set the force constant for this bond type to zero.
- Several previously hard-coded parameters are now user-setable preferences. Use the prefs command to list them and the set command to change.
- Force field parameters are loaded as a separate step using the readff command, not as part of the maketop command.
- The addbond command has new syntax and accepts input either as atom numbers or as residue_number:atom_name and will warn if the bond distance is large.
- The xlink command should be called before generating the topology, like the addbond command.
- Gap markers are not required between fragments without head or tail atom designation, *e.g.* not between solvent molecules.
- Overloading of library entries: Loading a library entry with the same name as a previously loaded entry removes the old definition.
- When loading a topology, all libraries used to make it will be loaded automatically.

D.8.2 Differences in qdyn

- Solvation and definition of the simulation sphere has been moved to Qprep.
- The keywords centre, radius, pack, model and water are obsolete and ignored.
- The number of atoms in the topology is always equal to the number of atoms in the simulation, *i.e.* atoms can not be added by reading a restart file.
- The CHARM DCD format is used for trajectory files. This file format can be read by visualisation programs like VMD [18] for trajectory animation.
- Shake can be enabled specifically for bonds to hydrogen atoms.
- Any tri-atomic molecule may be used as solvent, not only SPC or TIP3P water models.

D.9 Utility programs

A number of little programs and scripts that may be useful in the different steps in preparing or analyzing MD simulations are also available at the Q web site and briefly described here.

D.9.1 Qcalc

Command	Description
xscore	Scores topology, trajectories and restart files using the X-Score algorithm
chemscore	Scores topology, trajectories and restart files using the ChemScore algorithm
PMF-Score	Scores topology, trajectories and restart files using the PMF-Score algorithm

See section B.5 on page 31 for more information about scoring.

D.9.2 proq - PDB file analysis and manipulation

Proq is an interactive, shell-like program for examining and modifying pdb files prior to making topology or for analysis of structures from simulations. It is a non-graphical, command-oriented program written in PERL, intended to be a fast and powerful complement to molecular graphics programs. Proq can handle subsets of atoms, make distance matrices, calculate midpoints, change protonation states of residues, display charges, and calculate electrostatic interaction energies.

The contents of a PDB file will be loaded as fragments separated by comments, GAP marks etc. In the simplest case there will be only one fragment. The comments will be retained upon saving. Proq can not add or remove atoms. Change of protonation state is a matter of changing the residue name, *e.g.* from ASP to AS-.

A sample proq session is shown below, where user input is shown in bold face:

```
Hello! This is proq. Enter ? for help.

proq>proq>read : Reading input from h:\proq/ defaults.proq.
subset charged AR+:CZ,LY+:NZ,HIP:CE1,AS-:CG,GL-:CD
subset uncharged ARG:CZ,LYS:NZ,HIS:CE1,ASP:CG,GLU:CD
subset chargeable AR[G+]:CZ,LY[S+]:NZ,HI[SP]:CE1,AS[P-]:CG,GL[U-]:CD

charge AR+:CZ +1
charge LY+:NZ +1
charge HIP:CE1 +1
charge AS-:CG -1
charge GL-:CD -1

chargeform AR+ ARG
chargeform LY+ LYS
chargeform AS- ASP
chargeform GL- GLU

main : EOF reading h:\proq/defaults.proq.
main : Now reading from g:/config.proq again.
proq>
```

```
main : EOF reading g:/config.proq.
main : Now reading from STDIN again.
proq>load test
load : test.pdb consisting of 3011 atoms in 546 residues in 361 fragments.
proq>listsequence 3
listseq : sequence of fragments 3 of test.pdb
frg res type
3 189 MTX
proq> disttab charged MTX:N.*
B1:N1 B2:NA2 B3:N3 B4:NA4 B5:N5 B6:N8 B7:N10 B8:N
MTX189 MTX189 MTX189 MTX189 MTX189 MTX189 MTX189 MTX189
A 1:CG AS- 21: 11.40 13.72 12.98 12.54 9.86 9.10 7.08 11.62
A 2:CZ AR+ 28: 11.86 13.38 14.02 14.90 12.62 10.61 10.70 8.44
A 3:CD GL- 30: 3.62 3.48 5.42 7.56 7.21 5.06 9.21 9.63
A 4:CZ AR+ 32: 14.15 14.26 15.86 17.55 16.46 14.40 16.29 11.96
A 5:NZ LY+ 55: 16.46 18.30 16.50 14.83 13.56 14.79 12.43 18.48
A 6:NZ LY+ 68: 15.77 16.56 16.49 16.66 15.45 15.23 14.15 8.18
A 7:CZ AR+ 70: 11.09 11.36 10.78 10.63 10.50 11.22 10.82 6.15
A 8:CD GL- 172: 11.85 12.24 14.05 15.92 14.60 11.93 14.64 13.24
proq>midpoint MTX,charged
Computing midpoint for 52 atoms.
Point: 19.105 24.735 2.599 Max distance: 25.059
proq>centre C1 19.105 24.735 2.599
proq>disttab charged CTR
B1:C1
CTR 0
A 1:CG AS- 21: 23.42
A 2:CZ AR+ 28: 21.93
A 3:CD GL- 30: 18.89
A 4:CZ AR+ 32: 21.94
A 5:NZ LY+ 55: 25.06
A 6:NZ LY+ 68: 14.09
A 7:CZ AR+ 70: 8.05
A 8:CD GL- 172: 25.02
proq>off 21
chargeoff : Turned off charge on residue AS- 21.
proq>midpoint MTX,charged
Computing midpoint for 51 atoms.
Point: 20.556 21.556 0.018 Max distance: 22.745
proq>save test2
proq>quit
```

Table 14: proq command reference

Command, alias	**Arguments (optional)**	**Description**

Table 14: proq command reference

Command, alias	**Arguments (optional)**	**Description**
chargeonoff, onoff, onf	[r1[-r2]]	Flip the charged state of residues with numbers r1 through r2. *E.g.* all GLU will be renamed to GL- and vice versa.
centre, ctr	[centre_name [x y z]]	Add a reference point (like an atom) at (x,y,z). An atom named center_name is created in a pseudo-residue called CTR and numbered 0. Use show CTR to list all defined centers. Centers are not written with save.
charge, ch	[atom_descr. [charge]]	Define the electrostatic charge for all atoms that match atom_descr. atom_descr is one of residue_name:atom_name or residue_number:atom_name. When charges are evaluated, a residue_number:atom_name definition overrides a residue_name:atom_name definition. Regular expressions are not allowed.
chargeform, cf	[on_name [off_name]]	Define the residue names for the charged and neutral form of an amino acid. These pairs are used for switching charges on and off with chargeon and chargeoff. Ex. chargeform LY+ LYS. Charges must be assigned to atoms (*e.g.* LYS:NZ and LY+:NZ) for calculations with charges to work!
chargeon, on	[r1[-r2]] [f3....]	Switch names of the specified (by number) residues to their charged form. Only residues with a charged form defined are affected.
chdir, cd	[newdir]	Changes working directory.
clear		Clear the loaded PDB file from memory.
disttab, dist	[subset1 [subset2]]	Calculate distances between atoms in subsets 1 and 2 and display in a table. The number of atoms in subset2 should be <10 for neat screen output.
env	[option_name]	Display option settings. Default: all options.
help, h, ?	[command]	Display help for a command. Default is all commands.
listchargeform		List all pairs of charged and neutral residue names.
listfragment, lf	[f1[-f2]]	List residue numbers in specified fragments. Default: all fragments.
listpdb, lp		Same as save but writes to stdout.
listsequence, listseq, ls	[f1[-f2]] [f3....]	List residue names in specified fragments. Default: all fragments.
listsubset		List all subset definitions.
load, lo	[pdb_file]	Load a pdb file into memory. The extension .pdb may be omitted.

Table 14: proq command reference

Command, alias	**Arguments (optional)**	**Description**
loadrestart, lore	[restart_file]	Loads new coordinates for a previously loaded pdb file from a **qdyn** restart file. PDB file must have same number of atoms and same numbering as restart file, so use only pdb files with hydrogens created by Qprep.
midpoint, mid	[subset]	Calculate the midpoint (centre of the smallest possible that encompasses all atoms in the subset). Limited to 50 atoms.
option, o	[option_name] [value]	Set options.
print, p	[key]	Display the value for a key. Default is all keys.
quit		Quit proq (without saving anything).
read, r	[script_file]	Read commands from file. Will look for script_file and script_file.proq in all directories listed in option scriptpath.
readlib, rl	[library_file]	Read partial charges from Q library file.
repel	[subset1 [subset2]]	Calculate electrostatic potential between atoms in subsets 1 and 2 (Coulomb potential with e=80)
save, sa	[file_name]	Write pdb file from memory. The extension .pdb will be filled in by the program if omitted.
set, s	[key [value]]	Set a key to a value.
status, st		Display the number of fragments, residues and atoms loaded.
subset, sub	[name [atomset [,atom-set...]]]	Store a definition of a subset of atoms for later use. Atomsets are one of residue name, *e.g.* LYS residue name:atom name, *e.g.* LYS:NZ residue number, *e.g.* 13, residue number-residue number, *e.g.* 13-73, residue number:atom name, *e.g.* 74:CA, residue number-residue:atom name, *e.g.* 13-73:CA, name of existing subset, e,g. my_set name of existing subset:atom name Ex. my_set:N. Residue and atom names may be given as (PERL) regular expressions. For example AR[G+]:N.* gives all nitrogen atoms in ARG and AR+ residues. Note on regexps: Plus signs (+) are escaped by the program.
system, !	[command]	Execute operating system command, *e.g.* system ls*pdb.

Table 15: Further options for proq:

quiet:	suppress messages (if not blank or zero).
overwrite_warn:	ask before overwriting file when saving (if not blank or zero).
scriptpath:	semicolon-separated list of paths, in order of priority, to search for scripts.
editor:	name (and path if necessary) to editor called by edit command.
edit_in_background:	if set, launches editor in background and returns immediately (UNIX only!).
repel:	name (and path if necessary) of the repel program.
disttab:	name (and path if necessary) of the disttab program.
midpoint:	name (and path if necessary) of the midpoint program.

Table 16: mdsh - input preparation shell

function	Interactive, shell-like program for creating sequences of **qdyn** input files and a command file for running them. Mdsh can import existing input files, modify settings using keywords, generate sequences of input files with varying lambda values for perturbation simulations. It can also be used to easily set up a simulation by asking a series of questions. Mdsh script files are useful for managing standardized simulation protocols.
Input	interactive commands.
Output	interactive output,**qdyn** input files, command file (*e.g.* C shell script).
Usage	`mdsh`
notes	Enter `help` to show a list of commands.

Each element of the **qdyn** input file is associated with a keyword in mdsh. The set command changes the values. The keywords in mdsh are identical to those in the **qdyn** input file, except in the cases listed in the following table:

qdyn input file[files]	**mdsh**
topology	topologyfile
restart	restartfile
final	finalfile
trajectory	trajectoryfile
energy	energyfile
fep	fepfile
restraint	restraintfile

Data items may be more than one, like *e.g.* posrestraints. To enter a multi-line value, use backslash (\) at the end of a line to continue with more lines.

A mdsh session to set up a simple simulation is shown below. User input is shown in bold face.

```
mdsh>set basename test
set                          : Set key basename to test.
mdsh>set subname _1_
set                          : Set key subname to _1_.
```

```
mdsh>ask
ask                    : Enter values for parameters or press enter to cancel.
ask                    : set step 10000
ask                    : set temp 100
ask                    : set nb_update 50
ask                    : set rcpp 10
ask                    : set rcww 10
ask                    : set rcpw 10
ask                    : set rcq 99
ask                    : set shake_solvent on
ask                    : set topologyfile complex.top
ask                    : set trajectoryflag 25
ask                    : set energyflag 0
ask                    : set outputflag 10
ask                    : set fepfile complex.fep
ask                    : set lambdas 1
mdsh>make
mdsh>set temp 100
set                    : Set key temp to 100.
mdsh>make
mdsh>set temp 300
set                    : Set key temp to 300.
mdsh>make
mdsh>make
mdsh>make
mdsh>quit
```

mdsh commands

Command, alias	**Arguments (optional)**	**Description**
ask		Prompts for each missing required parameter (from check command).
Chdir, cd	[newdir]	Change working directory.
Check, c		Check that all necessary keys for make are set, report missing keys.
Edit	[file]	Launch and editor to edit a file.
Env	[option_name]	Display specified option(s). Default: all.
fake, f	[stepname]	Same as make but writes to standard output rather than to files.
help, h, ?	[command]	Display help for a command. Default: all commands.
make, m	[stepname]	Make a single input file and append commands to a comfile.

Command, alias	**Arguments (optional)**	**Description**
makelist, ml	[λ -list_file]	Make a series of input files (and command file entries) based on λ values in λ-list_file. The default λ-list_file is [subname].ll Current directory and all directories listed in option script_path are searched forλ-list files.
option, o	[option_name [value]]	Set options.
print, p	[key]	Display the value for a key. Default: all keys.
quit, q		Quit mdsh (without saving anything!).
read, r	[file]	Read commands from file.
reverselambdas, rl	[infile [outfile]]	Reverse contents of a lambda-list file (and strip empty lines).
set, s	[key [value]]	Set a key to a value.
system, !	[command]	Execute operating system command.ex.: ! ls*top
unset, u	[key]	Delete a key value.
writelambdas, wl	[file]	Create file with series of lambda values. Input is: λ1-start [λ2-start] [...] λ1-step [λ2-step] [...] λ1-end [λ2-end] [...]
inp2tab		Create a table from multiple input files.
tab2inp		Create multiple input files from a table. The heading of the first column is (the name of the) inputfile. All other headings must be keys in any order. This subroutine automatically adds the suffix .inp to the filenames in the first column.

Further options for mdsh

quiet:	Suppress messages (if not blank or zero).
scriptpath:	Semicolon-separated list of paths, in order of priority, to search for scripts in.
editor:	Name (& path if necessary) to editor called by edit command.
edit_in_background:	If set, launches editor in background and returns immediately (UNIX only).
md_program:	Which program to call in the command file. For each value of option md_program there should be a corresponding option entry containing the path to the program, *e.g.* if option md_program is qdyn then option qdyn must be set to the path to the executable.
inputformat:	The name of the PERL format (in mdsh_formats.pl) to use for writing input files.
comformat:	The name of the format to use for writing com files.

importformat:	The name of the subroutine (in mdsh_import.pl) that imports input files.
comfilepermission:	The (octal) file permission to set for com files (UNIX only). Note: you may use expressions of the form $options{optionname} for option values to access values of other options, or to modify the existing value. In fact, any valid PERL expression given in the option value will be evaluated.
Files:	The file config.mdsh in the same directory as the program is read at startup. Typically it will set the scriptpath option to the directory mdsh in the users home directory. The last command in config.mdsh is typically 'read defaults'. A personal script with default settings called defaults.mdsh and located in [home dir]/mdsh/ will thus automatically be read at startup.

D.9.3 Analysis programs

Qave

function:	Calculate average energies and RMS deviations from **qdyn** log files.
input:	**qdyn** log files.
output:	Energy summary tables like the ones in the log file with average energies and RMS deviations on stdout.

usage:	`Qave [-skip n_skip]logfiles`
	`logfiles: Names of one or more \textbf {qdyn} log files-s,-skip`
	`Ingore the first n_skip energy summaries`

Qavetr

function:	Calculate average coordinates and RMS deviations from trajectories.
input:	**qdyn** trajectory files. The program will prompt for file names.
output:	Co-ordinate file (same format as restart file) with average coordinates and RMS deviations (instead of velocities in restart file).
usage:	`Qavetr` Enter names of trajectory files, one per line. End with bland line. Then enter the name of the co-ordinate file to be created.

lsextr

function:	Extract ligand-surrounding interaction energies from **qdyn** log files.
input:	**qdyn** log files.
output:	Q-atom-surrounding interaction energies to stdout: V_LJ V_el.
usage:	`lsextr logfiles` `logfiles: Names of one or more \textbf {qdyn} log files`

ineff

function:	Compute statistical inefficiency of a time series of correlated data while dividing the series into blocks of increasing size. Use the output to plot the inefficiency vs the block size, extrapolate to infinity and use to approximate the error of the mean.
input:	text file, optionally multiple columns.
output:	summary to stderr and table of inefficiency values to stdout:n_blocks points_per_block 1/points_per_block variance ineffsummary to stderr.

usage:	`ineff [-q] [filename] [column] [skip] [read]`
	`filename: text file with columns of (white space-separated) numeric data`
	`column: column to work on (1 for first)`
	`skip: number of lines to skip at beginning of file`
	`read: number of lines (data points) to read`
	`-q: suppress table output on stdout`
	`All arguments are optional. The program will prompt for missing arguments.`

tstart

function:	Find a starting point in a time series after which the average value is stable.
input:	text file, optionally multiple columns.
output:	

usage:	`tstart [-q] [filename] [column] [skip] [read]`
	`filename: text file with columns of (white space-separated) numeric data`
	`column: column to work on (1 for first)`
	`skip: number of lines to skip at beginning of file`
	`read: number of lines (data points) to read`
	`-q: suppress table output on stdout`
	`All arguments are optional. The program will prompt for missing arguments.`

D.9.4 Other utility programs

mkfep

function:	Create a basic FEP file with a list of Q-atoms.
input:	Topology numbers of first and last atom to be designated as q-atoms (on command line).
output:	FEP file.
usage:	`mkfep first last > simple.fep`

bone

function:	Compress log files, can be used as a filter to compress **qdyn** output on the fly. Bone separates numbers (flesh) from lines of text and stores the numbers in a binary file and the text templates (bones) in a template library file. Compression ratio is typically 1:5.
input:	**qdyn** log files (or any text file with repeated similar lines).
output:	Binary data file, template library file.

usage:	`\textbf {qdyn} test.inp | bone > test.log.bin`
	or
	`bone uncompressed.log`

notes:	Use bone -h to get information on options. Never remove the library file (bonelib.dat) or you won't be able to decompress!

enob

function:	The reverse of bone: decompress log files. Enob can also extract selected lines, *e.g.* Q-atom energies, from compressed files.
input:	Files compressed with bone, template library file.
output:	Text file (to stdout).

usage:	`enobtest.log.bin | more`
	or
	`enob -u compressed.bin`

notes:	Use enob -h to get information on options.

re2pdb

function:	Easily convert restart files to PDB files.
input:	topology file name, restart file name(s) on command line.
output:	PDB files.
usage:	`re2pdb molecule.top *re`
notes	Use re2pdb -h to get information on options.

Qdum

function:	Test **qdyn** input files quickly.
input:	**qdyn** input file.
output:	Log file, restart file.
usage:	`Qdum4 test.inp`
notes:	Qdum is a version of **qdyn** without the dynamics loop. It reads all input, initiates the simulation, writes a restart file and terminates. Qdum is build from the same source code as **qdyn**.

E SYSTEM GUIDE

E.1 System requirements

The Q programs will run under many operating systems including MS Windows NT/XP, Digital UNIX, LINUX, SGI IRIX, Sun Solaris, Cray UNICOS. A parallel version using the Message Passing Interface (MPI) standard is included in Q 5.0.

The memory requirements vary with the size of the simulated system but are, in general, modest. A system of 18 Å radius with 10 Å non-bonded cut-off takes about 3 to 7 Mb

depending on the computer and operating system.

E.1.1 Compilers

A Fortran90 compiler is required to build the main programs (**qdyn**, Qprep5, Qfep5, Qdum5, Qcalc5).

A Fortran90 compiler and a MPI library is required to build the parallel version of **qdyn**.

A Fortran77 compiler is required to build some helper programs called by proq (disttab, repel, midpoint).

A C compiler is required to build some of the analysis programs (tstart, ineff).

A PERL interpreter is required to run most of the utility programs (bone, enob, lsextr, mdsh, mkfep, proq, Qave).

E.2 Installation

The following components, available from the Q web site http://xray.bmc.uu.se/ aqwww/Q, are required to use Q:

1. Either executable images (available for some popular platforms) or source code (to compile on your own system).
2. Force field files.
3. Solvent files.

Make a directory such as ∼/Q, /usr/local/Q or C:\Q and download the files to this directory.

E.2.1 Installing executable images

MS Windows systems

1. Create a subdirectory in the Q directory for the programs, such as C:\Q\bin.
2. Uncompress the Pkzip archive file containing the windows executables to this directory using *e.g.* unzip or WinZip.
3. Add this directory to your search path by modifying the PATH environment variable in the system properties control panel.

UNIX system

1. Create a subdirectory in the Q directory for the executables, such as Q/bin.
2. Extract the executables from the gzip compressed tar archive file into this directory, *e.g.*
   ```
   gunzip -c Q_DigitalUNIX4_Alpha21164.tar.gz | (cd bin; tar -xvf -)
   ```

3. Add the bin directory to your search path by modifying the PATH environment variable (typically in ∼/.cshrc)

E.2.2 Building the programs from source code

MS Windows systems Project files and workspace file for Digital Visual Fortran v.6 are provided in the Windows source code distribution. After extracting the files from the zip archive, open the workspace file and build the programs using the configurations (Alpha or x86) that match your system.

UNIX systems The makefile included in the UNIX source code distribution contains settings (compiler flags) for some popular platforms including Compaq Tru64 UNIX, Linux and SGI Irix.

1. Create a subdirectory in the Q directory for the executables, such as Q/bin.
2. Create a subdirectory in the Q directory for the source files, such as Q/src.
3. Extract the source files from the gzip compressed tar archive file into this directory, *e.g.*
   `gunzip -c Q4_src.tar.gz |(cd src; tar -xvf -).`
4. Type make in the source directory. This will list the platforms supported in the makefile.
5. (a) If your platform is on the list, enter the appropriate make command, *e.g.*
   `make alpha-osf1-ev6`.
   (b) Please note that the compiler optimisation settings in the makefile are in no way guaranteed to produce the fastest code. Feel free to try different settings and if you can improve the result, please notify us.
6. (a) If your platform in not on the list, you need to find the appropriate compiler flags yourself and edit the makefile accordingly. Consult your F90 manual. Use the generic settings as s starting point:
   `make generic`.
   (b) When you have updated the make file for your system, please e-mail it to us, and we will merge your changes into the distributed makefile.
7. Move the executables to the bin directory.
8. Add the bin directory to your search path by modifying the PATH environment variable (typically in ∼/.cshrc).

References

[1] Marelius, J., Kolmodin, K., Feierberg, I. and Åqvist, J. Q: An MD program for free energy calculations and empirical valence bond simulations in biomolecular systems, J. Mol. Graphics Mod. 1999, 16, 213-225.

[2] Kollman, P.A., Free Energy Calculations: Applications to Chemical and Biochemical Phenomena. Chem. Rev. 1993, 2395-2417.

[3] Beveridge, D.L. and DiCapua, F.M., Free energy via molecular simulation: applications to chemical and biomolecular systems. Annu. Rev. Biophys. Biophys. Chem. 1989, 431-492.

[4] Warshel, A., Computer modeling of Chemical reactions in Enzymes and Solutions. Wiley, New York, 1988.

[5] Åqvist, J. and Warshel, A., Simulation of Enzyme Reactions Using Valence Bond Force Fields and Other Hybrid Quantum/Classical Approaches. Chem. Rev. 1993,93,2523-2544.

[6] Åqvist, J., Medina, C. and Samuelsson, J.E., A new method for predicting binding affinity in computer-aided drug design., Protein Engng. 1994, 7, 385-391.

[7] Jones-Hertzog, D.K. and Jorgensen, W.L., Binding Affinities for Sulfonamide Inhibitors with Human Thrombin Using Monte Carlo Simulations with a Linear Response Method. J. Med. Chem. 1997, 40, 1539-1550.

[8] Hansson, T., Marelius, J. and Åqvist, J., Ligand Binding Affinity Prediction by Linear Interaction Energy Methods, J. Comput.-Aided Mol. Des. 1998, 12, 27-35.

[9] Warshel, A., A microscopic model for calculations of chemical processes in aqueous solution, Chem. Phys. Lett. 1978, 55, 454-458.

[10] Berkowitz, M. and McCammon, J.A., Molecular Dynamics with Stochastic Boundary Conditions, Chem. Phys. Lett. 1982, 90, 215-217.

[11] Brünger, A. T., Brooks, C.L. III. and Karplus, M., Stochastic Boundary Conditions for Molecular Dynamics Simulations of ST2 Water, Chem. Phys. Lett. 1984, 105, 495-500.

[12] See *e.g.* http://www.rcsb.org/pdb/docs/format/pdbguide2.2/guide2.2_frame.html.

[13] Berendsen, H.J.C., Postma, J.P.M., van Gunsteren, W.F. and Hermans, J., Interaction models for water in relation to protein hydration, In Intermolcular Forces, Pullman, B. (ed.). Riedel, Dordrect, 1981, pp 331-342.

[14] Jorgensen, W.L., Chandrasekhar, J., Madura, J.D., Impey, R.W. and Klein, M.L., Comparison of Simple Potential Functions for Simulating Liquid Water. J. Chem. Phys. 1983, 79, 926-935.

[15] Kolmodin, K. and Åqvist, J, PTP proton transfer...

[16] Åqvist, J., Free Energy Perturbation Study of Metal Ion Catalyzed Proton Transfer in Water. J. Phys. Chem. 1991, 95, 4587-4590.

[17] http://www.ibg.uu.se/upload/PetraW.pdf Wennerström, P. (valid link 2003-06).

[18] Humphrey, W., Dalke, A. and Schulten, K., VMD -Visual Molecular Dynamics, J. Mol. Graph., 1996, 14, 33-38.

[19] VMD home page:http://www.ks.uiuc.edu/Research/vmd/ Laaksonen, L., A (valid link 2003-06).

[20] graphics program for the analysis and display of moleculardynamics trajectories, J. Mol. Graph. 1992, 10, 33.

[21] gOpenMol home page: http://www.csc.fi/ laaksone/gopenmol/gopenmol.html valid link (2003-06).

[22] Cornell, W.D., Cieplak, P., Bayly, C.I., Gould, I.R., Merz, K.M. Jr., Ferguson,D.M., Spellmeyer, D.C., Fox, T., Caldwell, J.W. and Kollman, P.A., A second generation force field for the simulation of proteins and nucleic acids. J. Am. Chem. Soc. 1995, 117, 5179-5197.

[23] Jorgensen, W.L. and Tirado-Rives, J., The OPLS Potential Functions for Proteins.Energy Minimizations for Crystals of Cyclic Peptides and Crambin. J. Am. Chem. Soc. 1988, 110, 1657-1666.

[24] Brooks, B.R., Bruccoleri, R.E., Olafson, B.D., States, D.J., Swaminathan, S. and Karplus, M., CHARMM: A Program for Macromolecular Energy, Minimization, and Dynamics Calculations. J. Comp. Chem. 1993, 4, 187-217.

[25] van Gunsteren, W.F. and Berendsen, H.J.C., Groningen Molecular Simulation (GROMOS) Library Manual. Biomos B.V., Groningen, 1987.

[26] van Gunsteren, W. F., [GROMOS87 modification].

[27] van Gunsteren, W.F., Billeter, S.R., Eising, A.A., Hünenberger, P.H., Krüger, P., Mark, A.E. and Tironi, I.G., Biomolecular Simulation: The GROMOS96 Manual and User Guide. Hochschulverlag AG an der ETH Zürich, 1996.

[28] Jorgensen W.L., Maxwell D.S., Tirado-Rives J., Development and Testing of the OPLS All-Atom Force Field on Conformational Energetics and Properties of Organic Liquids. J.Am.Chem. Soc. 1996, 118, 11225-11236.

[29] Essex, J.W. and Jorgensen, W.L., An Empirical Boundary Potential for Water Droplet Simulations. J. Comp. Chem. 1995, 16, 951-972

[30] Berendsen, H.J.C, Postna, J.P.M., van Gunsteren, W.F., di Nola, A. and Haak, J.R., Molecular dynamics with coupling to an external bath, J. Chem. Phus. 1984, 81, 3684-3690.

[31] Zacharias M., Straatsma T.P., McCammon J.A., Separation-shifted scaling, a new scaling method for Lennard-Jones interactions in thermodynamic integration. J.Chem.Phys. 1994, 100, 9025-9031.

[32] Renxiao Wang, Luhua Lai and Shaomeng Wang, Further development and validation of empirical scoring functions for structure-based binding affinity prediction, Journal of Computer-Aided Molecular Design, 16, 11-26, 2002

[33] Matthew D. Eldridge, Christopher W. Murray, Timothy R. Auton, Gaia V. Paolini and Roger P. Mee, Empirical scoring functions: I. The deveoplment of a fast empirical scoring function to estimate the binding affinity of ligands in receptor complexes, Journal of Computer-Aided Molecular Design, 11, 425-445, 1997

[34] Ingo Muegge and Yvonne C. Martin, A general and fast scoring function for protein–ligand interaction: A simplified approach, J. Med. Chem., 42, 701-894, 1999