PB - a Poisson-Boltzmann equation solver

Andreas Kerzmann

June 2, 2014

This document briefly describes the usage of the `PB` application.

1 Building and using `PB`

If you did not build the BALL library yet, read the tutorial and install the BALL library first.

Change to the directory where you stored the BALL source files, say `/opt/BALL/source`. Then change to the directory of the `PB` application `APPLICATIONS/PB`. Issue the `make` command to build an executable named `PB`, which is the program this short introduction is all about. In short:

```
cd /opt/BALL/source
cd APPLICATIONS/PB
make
```

Now that the executable is built, we can start our first test run. Make sure your `LD_LIBRARY_PATH` environment variable is set correctly. If you do not know how to do this or what I am talking about, please refer to the BALL tutorial. Run the program by typing `./PB` and pressing the Enter-key. The output should look like figure 1.

This usage message will always appear when starting the application without giving any options. As you can see, many calculation options can be controlled via the command line interface. For further options you can use a so-called `options` file containing more (and less frequently used) options to tailor the calculation according to your needs. The usage of the `options` file will be briefly explained later.

2 A simple expample

Imagine you want to calculate the solvation free energy of a simple monovalent spherical ion in pure water (*i. e.* no salt solution with counter ions), say

```
BALL -- Finite Difference Poisson Solver

PB [<options>]
   where <options> is one or more of the following possibilities:
     -P                   perform a Finite Difference Poisson calculation
     -A                   calculate the solvent accessible surface and
                            volume of the solute
     -E                   calculate the solvent excluded surface and
                            volume of the solute

further options:
     -p <FILE>            read <FILE> as a PDB file
     -h <FILE>            read <FILE> as a HyperChem file
     -H <FILE>            read <FILE> as a HyperChem file but do not assign
                            charges
     -o <FILE>            read FDPB options from <FILE>
     -c <FILE>            read charges from <FILE>
     -r <FILE>            read radii from <FILE>
     -t <FILE>            read charge and radius rules from <FILE>
     -u <FILE>            read charge rules from <FILE>
     -w <FILE>            read radius rules from <FILE>
     -0                   clear all charges in subsequently read structures
     -s                   calculate the solvation free energy by performing a
                            second FDPB calculation in vacuum
     -n                   normalize all atom names in subsequently read
                            structures
     -b                   try to build the bonds (e.g. for PDB files)
     -d <FILE>            dump the atom charges, radii, and surface fractions
                            to <FILE> (for debugging)
     -v                   verbose output (implies ``verbosity 99'' in the
                            option file, print additional results and options)
     -x <RADIUS>          the probe sphere radius for solvent accessible and
                            solvent excluded surface calculations
                            [default = 1.4 A]
     -e <DIEL_CONST>      the dielectric constant of the surrounding medium
                            [default = 78.0]
     -f <DIEL_CONST>      the dielectric constant of interior of the solute
                            [default = 2.0]
     -i <IONIC_STRENGTH>  the ionic strength which will be used for the
                            Boltzmann part of the Poisson-Boltzmann equation
                            [default = 0.0 mol/l, i. e.  switched off]

  By default, charges and radii are taken from data/charges/PARSE.crg
  and data/radii/PARSE.siz.

  Charge and radius assignment options can be repeated. They are valid for all
  subsequently read structures.
```

Figure 1: Help message of PB

K^+, which you have stored in a HyperChem HIN file named `k.hin`. Additionally you want to compare the calculated value with the Born approximation of the electrostatic contribution to solvation free energy.

To perform the calculation you need a charge and a radius for the ion. As you stored the ion in a HyperChem file, its charge should already be contained there. The radius has to be assigned by using another file containing rules for assigning radii to atoms. For simplicity of the example this data is stored in the file `ions.rul`. See the BALL reference manual for the usage of these rule files.

As you can see from the usage message (see figure 1) you will need the following options for the calculation:

-P perform a Poisson calculation

-s calculate the solvation free energy

-w read radius rule files from the file which is specified after this option

-v be verbose (this is optional but try it once to see which information you can get from this option)

-h use a HyperChem HIN file as input (specify the file name immediately after this option)

Compiling all those options together will create the actual command which you will have to type:

```
./PB -P -s -w ions.rul -h k.hin
```

If the output looks somewhat similar to figure 2 then you successfully calculated your first solvation free energy with the BALL FDPB solver, congratulations!

```
[18:29:16] Calculating the solvation free energy.
[18:29:16] first calculation step: solvent dielectric constant = 78.000000
[18:29:16] total energy:          28.8077 kJ/mol
[18:29:16] reaction field energy: -243.647 kJ/mol
[18:29:16] Calculating the solvation free energy.
[18:29:16] first calculation step: solvent dielectric constant = 78.000000
[18:29:16] second calculation step: solvent dielectric constant = 1.0 (vacuum)
[18:29:16] total energy:          383.598 kJ/mol
[18:29:16] reaction field energy: 99.2415 kJ/mol
[18:29:16]
[18:29:16] Solvation energy as change of the total energy:   -354.79 kJ/mol
[18:29:16] Solvation energy as change of the reaction field: -342.889 kJ/mol
```

Figure 2: Output of a simple ion solvation calculation

Now we check against the Born[?] formula (equation 1) for calculating the solvation free energy of spherical ions in water.

$$\Delta G_{\text{solv}} = \frac{-z^2}{r_i}\left(1 - \frac{1}{\varepsilon_s}\right)\frac{e_0^2 N_A}{8\pi\varepsilon_0} \tag{1}$$

Inserting our values for the K^+ ion (charge of 1, radius of 2.0Å) yields a solvation free energy of -342.80 kJ/mol which is in good agreement with our calculated values.

3 The `options` file

The Poisson-Boltzmann solver of BALL can be controlled via many options. The most relevant options will be introduced in this short introduction. Please consult the BALL reference manula for documentation concerning further options. There is a commented `options` file `PB.options` in the source directory of the `PB` application which contains all tunable options. This file will also be in the appendix.

The syntax of the options file is simple. It consists of lines which start with an option keyword and contain the option value as the rest of the line. Lines beginning with `;` or `#` will be ignored and can be used for comments. Note that option keywords must not contain whitespaces, *i.* e. blanks, tabs and so on. Empty lines are also forbidden.

```
; This is an example for the options syntax
;
; an imaginary option setting the number of hobbits
hobbit_count 4
```

3.1 Options controlling grid properties

spacing is an option controlling the grid granularity. It defines the distance in Ångstrom (Å) between every two grid points of the same axis-parallel line.

border sets the minimum distance in Å between the bounding box of the molecule and the border of the grid.

3.2 Options controlling the solver

boundary_condition sets the method for defining the boundary condition of the equation system. There are five different methods from which you can choose:

zero simply sets potential to zero at the border

Debye uses the Debye approximation for the boundary

Coulomb *not implemented at the time of writing*

dipole *(default)* uses the effective dipole as boundary condition

focusing computes a larger but less dense grid and takes its result for inititalizing the original grid. This method gives most accurate results but takes a lot more computing time.

charge_distribution controles the distribution of molecule charges over grid points. You can choose between two algorithms.

uniform *(default)* uses a uniform distribution scheme for assigning charges to grid points

trilinear distributes charges equally upon the eight closest grid points

dielectric_smoothing activates a smoothing algorithm for the dielectric constant. There are three options avilable.

none switches dielectric smoothing off

uniform uses a uniform smoothing function

harmonic uses a harmonic smoothing function

3.3 Options controlling the molecular environment

These options are also adaptable through command line switches

ionic_strength sets the ionic strength of the sourrounding medium. It has to be specified in mol/l. Command line switch: -i.

solvent_dielectric_constant defines the dielectric constant of the continuous solvent. Command line switch: -e.

solute_dielectric_constant defines the dielectric constant of the solute molecule(s). Command line switch: -f.

A The PB.options file

```
; $Id: PB.tex,v 1.3 2002/12/16 14:34:43 anker Exp $
; Example options file for the PB application
;
; all options are commented out
; just remove the ';' at the beginning of the options in order to use them
;
; set the verbosity level of the application
; 0 produces very little output while 99 creates a lot of information
;verbosity 10
;
; print the timing of all computationally expensive functions
;print_timing true
;
; set the grid spacing, i. e. the distance between every two grid points of
; the same axis-parralel line (in Angstrom)
;spacing 0.5
;
; set the border of the system, i. e. the space between the bounding box of
; the molecule(s) and the edge of the grid (given in Angstrom)
;border 5.0
;
; define the ionic strength of the solvent, i. e. activate the Boltzmann
; part of the calculation
; use units of mol/l
; this option can also be set by using the -i command line switch
;ionic_strength 0.15
;
; set the dielectric constant of the solute
; usually 2.0 is the value used in protein calculations
;solute_dielectric_constant 4
;
; set the solvent dielectric constant
; water has a dielectric constant of 78.5
;solvent_dielectric_constant 78.5
;
; set the probe radius for SAS calculations
; one usually assumes 1.4 Angstrom as the "radius" of water molecules
;probe_radius 1.4
;
; set the ion radius, i. e. the space around the molecule where ions from
; the solvent are exluded (in Angstrom)
;ion_radius 2.0
;
; set the temperature of the system (in Kelvin)
; the default is 298.15 Kelvin
;temperature 290
;
; define the boundary condition for inititalizing the equation solver
; there are several options: zero, Debye, Coulomb, dipole and focusing
;boundary_condition focusing
;
; choose the algorithm for distributing the atom charges over the grid
; there are two methods: uniform and trilinear
;charge_distribution uniform
;
; choose the method of smoothing the dielectric constant
; there are three methods: none, uniform and harmonic
; note that reaction field energies cannot be computed if dielectric
; smoothing is activated
```

```
;dielectric_smoothing none
;
; define a spatial offset for the calculation
; you can displace the whole grid by defining this offset vector
;offset (1.0 1.0 1.0)
;
; set the rms criterion for termination of the solver iterations
; if the root mean sqaure deviation of the equation system is below this
; value and the maximum resiual is below the max_criterion limit the
; equation system did converge
;rms_criterion 1e-6
;
; set the maximum residuals criterion for the solver
; if the root mean sqaure deviation of the equation system is below this
; value and the maximum resiual is below the max_criterion limit the
; equation system did converge
;max_criterion 1e-6
;
; set the frequencz of checking the convergence criterions
; use this options for defining the number of iterations that will be
; performed before checking the criterions again.
;check_after_iterations 5
;
; set the maximum number of iterations
; if the solver did not converge until thid number of iterations stop the
; solver
;max_iterations 1000
;
; set the lower corner of the grid
;lower (0.0 0.0 0.0)
;
; set the upper corner of the grid
;upper (0.0 0.0 0.0)
;
; set the lower corner of the bounding box of the molecule(s)
;bounding_box_lower (0.0 0.0 0.0)
;
; set the upper corner of the bounding box of the molecule(s)
;bounding_box_upper (0.0 0.0 0.0)
```