BornProfiler Documentation

Release 1.0.0

Lennard van der Feltz, Kaihsu Tai, Oliver Beckstein

Sep 03, 2018

Contents

BornProfiler is a Python package to set up calculations of the electrostatic free energy of an ion in a membrane protein for the Poisson-Boltzmann solver APBS.

CHAPTER 1

Contents

1.1 Overview

BornProfiler is a collection of scripts to set up Poisson-Boltzmann electrostatic calculations for the APBS package, in particular calculations of the electrostatic solvation free energy of an ion along a pathway in a membrane protein (the so-called *Born profile*).

1.1.1 Features

The BornProfiler package helps setting up Poisson-Boltzmann calculations of the electrostatic potential of mean force of an ion in a pore or channel under the influence of a membrane. The membrane is modelled as a dielectric slab of epsilon=2.

- Provide a path (list of coordinates) and a PQR file of the protein as input.
- A membrane can be defined with arbitrary thickness, z-position, and dielectric. A headgroup region can also be defined with a different dielectric constant.
- Define all input parameters in a compact parameter file so that there is always a record of the exact calculation setup available.
- Born radii for all ions from the Rashin & Honig paper *[Rashin1985]* are included; just select the ion in the input file.
- Born radii for H3O+, OH- (and H+... for testing) have been derived from the solvation free energies in *[Pliego2000]* directly via the Born equation. USE AT YOUR OWN RISK!!
- Customize run scripts and queuing system submission scripts by providing your own templates.

1.1.2 History and Contributions

Based on Kaihsu Tai's Python rewrite (Poisson-Boltzmann profile for an ion channel) of the original `placeion.sh` and `analyze.sh` bash scripts by Kaihsu Tai and Oliver Beckstein.

Uses material from the APBS Wiki (PMF of a helix in a membrane) and contains a modified version of Michael Grabe's `draw_membrane2` from APBSmem.

1.2 Building and installing BornProfiler

BornProfiler consists of a Python package `bornprofiler` and a a stand-alone executable `draw_membrane2`. The Python package is needed to set up the calculations and `draw_membrane2` is needed as a helper tool for `apbs`; hence it needs to be installed on the same machine where `apbs` is going to run.

1.2.1 Required pre-requisites

- python2.5 or better
- NumPy
- a C compiler such as GNU gcc
- APBS >= 1.3

1.2.2 Installation from Source

Unpack the tar ball:

```
tar zxvf BornProfiler-1.0.tar.gz
```

Install the python module and scripts:

```
cd BornProfiler
python setup.py install --user
```

(`--user` might only work for Python 2.6; look at the output of `python setup.py install --help` for guidance on what your options are.)

Compile the customized (and improved) version of `draw_membrane` named `draw_membrane2`:

```
mkdir BUILD && cd BUILD
cmake -D CMAKE_INSTALL_PREFIX=$HOME -D CMAKE_BUILD_TYPE=Release ../src/drawmembrane
make
make install
```

The `make install` step will install the executable `draw_membrane2a` under `CMAKE_INSTALL_PREFIX/bin`; change `CMAKE_INSTALL_PREFIX` if you prefer another location.

(cmake is not really needed; if you don't have it try the following:

```
gcc ./src/drawmembrane/draw_membrane2a.c -o draw_membrane2a -lm -lz
```

and install manually in a place where you or your shell can find it.)

Note: `draw_membrane2a` also needs to be installed on the machine where you want to run your BornProfiler jobs: it will run together with `apbs`. If you are going to run you calculations on a cluster then `draw_membrane2a` (and `apbs`) need to be *both* installed on the cluster.

1.2.3 Configuration

Finalize your installation by running

```
apbs-bornprofile-init.py
```

This should tell you that it set up a configuration file `~/.bornprofiler.cfg` and a number of directories.

The default `~/.bornprofiler.cfg` looks like this:

```
[DEFAULT]
configdir = ~/.bornprofiler
templatesdir = %(configdir)s/templates
qscriptdir = %(configdir)s/qscripts

[executables]
apbs = apbs
drawmembrane = draw_membrane2a
```

The file can be edited in a text editor. For instance, one can add the full path to the `apbs` and `draw_membrane2a` executable binaries.

Any other variables used in run configuration input files can also be added here and will be used as defaults.

Advanced use: You can drop templates for run scripts into *qscriptdir* and have the BornProfiler package pick them up automagically.

1.3 User documentation

This documentation will tell you what you can achieve with **BornProfiler**, sketch out the background, what the typical workflow looks like, and discuss some examples.

1.3.1 A simple Born profile

TODO: Outline the problem of ion permeation, discuss simple example and show how this package can solve the problem. Choose something very simple such as nAChR or GLIC.

1.3.2 Background

TODO: Describe the theoretical background and say how individual steps are done in the package.

1.3.3 Workflow

TODO: describe the individual steps

Sample point generation

Path (1D)

- str8path
- HOLE

Volume (3D)

- HOLLOW (custom version)

Parameter settings

Radii and Charges

Use pdb2pqr to generate the input PQR file.

BornProfiler run input file

- ...
- membrane position
- exclusion zone

Queuing system script

Discuss example, highlight what needs to be customized and how.

Generate window input files

:: apbs-bornprofile-mplaceion

Run jobs

Manually or typically through a queuing system.

Analyze and visualize data

:: apbs-bornprofile-analyze

Distinguish 1D/3D.

Talk about using Chimera to analyze 3D energy landscapes and add some tips & tricks.

1.3.4 Examples

Two examples are part of the BornProfiler source distribution.

Parsegian

Nicotinic acetylcholine receptor (nAChR)

1.4 Developer documentation

Most users will likely use **BornProfiler** through the provided scripts. However, in order to extend functionality and develop new tools one can also use the `bornprofiler` Python module as a library.

Content:

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Bibliography

[Rashin1985] A.Rashin & B.Honig, J Phys Chem B 89 (1985), 5588

[Pliego2000] J.R. Pliego and J.M. Riveros. Chemical Physics Letters, 332(5-6): 597–602, 2000. doi:10.1016/S0009-2614(00)01305-1.