Smolfin documentation

Peter Kekenes-Huskey (pkekeneshuskey at ucsd.edu)

Smolfin provides a numerical solution to the Smoluchowski equation, which models the diffusion of a ligand subject to an electrostatic PMF. Here we solve for the steady-state distribution of Ca2+, using a realistic representation of a protein geometry [**?**].

In this tutorial, we produce a finite element mesh of a protein, determine the electrostatic potential about the protein, and numerically solve the Smoluchowski equation.

Organization:

1. a short tutorial Section 1
2. Theory Section 2
3. Mesh generation Section 3
4. Electrostatic potential generation Section 4
5. Running smol Section 5
6. Analyzing results Section 6
7. Installation Section 8

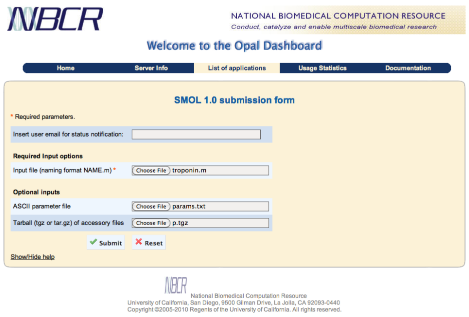
1 The short tutorial:

1.1 Prep

- Follow Gamer prep described in manual (Section 3) to generate .m file (or download example file at http://mccammon.ucsd.edu/~huskeypm/share/workshop/troponin.m)
- Follow APBS prep described in manual (Section 4) to generate .dx files (or download example http://mccammon.ucsd.edu/~huskeypm/share/workshop/p.tgz)

1.2 Run

- navigate to http://rocce-vm6.ucsd.edu/opal2/CreateSubmissionForm.do?serviceURL=http://localhost:8080/opal2/services%2Fsmol_1.0


NBCR
NATIONAL BIOMEDICAL COMPUTATION RESOURCE
Conduct, catalyze and enable multiscale biomedical research
Welcome to the Opal Dashboard
Home
Server Info
List of applications
Usage Statistics
Documentation

SMOL 1.0 submission form

* Required parameters.

Insert user email for status notification:

Required Input options

Input file (naming format NAME.m) *: troponin.m

Optional inputs

ASCII parameter file: params.txt

Tarball (tgz or tar.gz) of accessory files: p.tgz

Submit
Reset
Show/Hide help
NBCR
National Biomedical Computation Resource
University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093-0440
Copyright ©2005-2010 Regents of the University of California. All rights reserved.

Figure 1: Screen shot of Smolfin interface with files for uploading.

- Upload `${VAR}.m` file from MolecularMesh
- Optional: To include electrostatic potentials form APBS,

  1. upload parameter file with

     ```
     D 1
     q 1
     dx apbs-LowestResolution.dx
     ...
     dx apbs-HighestResolution.dx
     ```

     Here D [$\mu m^2/ms$] is the diffusion constant, q [e] is the charge, and dx is an apbs file with the electrostatic potential [kT/e]. (see example file at http://mccammon.ucsd.edu/~huskeypm/share/workshop/params.txt). **It is imperative that the .dx files are listed in the parameter file in the order of increasing resolution.**

    2. upload tarball of dx files

- Click submit.

1.3 Analyze

- After a few minutes, the Opal service will guide you to an output directory with your files. There's a lot printed to the directory, but the most important files are packaged in a file called `${VAR}_outputs.tgz.`
- the following .pvd/.vtu files can be visualized in paraview (Section 6)
    - `${VAR}_values.pvd/.vtu` - paraview files for electrostatic potential
    - outputs/up.pvd/.vtu - steady-state solution for Ca2+ about troponin

```
D 1
q 1
dx potential-15-PE0.dx
dx potential-14-PE0.dx
dx potential-13-PE0.dx
dx potential-12-PE0.dx
dx potential-11-PE0.dx
dx potential-10-PE0.dx
dx potential-09-PE0.dx
dx potential-08-PE0.dx
dx potential-07-PE0.dx
dx potential-06-PE0.dx
dx potential-05-PE0.dx
dx potential-04-PE0.dx
dx potential-03-PE0.dx
dx potential-02-PE0.dx
dx potential-01-PE0.dx
dx potential-00-PE0.dx
```

2 Theory

The time-dependent Smoluchowski problem is formulated as follows: Find an ionic concentration function $\rho : OL\Omega_e \times [0, T] \rightarrow R$ such that

$$\frac{\partial \rho}{\partial t} = D(\Delta \rho - \nabla \cdot (\beta \rho \ \vec{E})) \tag{1}$$

in Ω_e, where D is the diffusion constant, $\beta := 1/k_B T$ is the inverse Boltzmann energy with k_B the Boltzmann constant and T the temperature, and $\vec{E}$ is the electric field derived from the potential $\phi(x)$ on Ω_e via the standard equation $\vec{E} := -\nabla \phi$. The boundary conditions are taken to be

$$\vec{J} \cdot \hat{n} = 0 \quad \text{on } \Gamma_s, \qquad \rho = \alpha \quad \text{on } \Gamma_a, \qquad \rho = \rho_b \quad \text{on } \Gamma_e, \tag{2}$$

where $\vec{J} := -D(\nabla \rho - \beta \rho \vec{E})$ is the flux of ρ, α is the relative absorption of substrate at the active site, and ρ_b is the bulk ionic concentration, assumed to be constant outside of Ω_e, and $\hat{n}$ is the surface normal along Γ. To compute the reaction rate in this context, one solves the steady state problem

$$\frac{\partial \rho}{\partial t} = D(\Delta \rho - \nabla \cdot (\beta \rho \ \vec{E})) = 0, \tag{3}$$

and computes the accompanying steady state flux $\vec{J}_{ss}$. The reaction rate for the steady state reaction rate with no channel (e.g. the diffusional encounter), ka is then determined by an area integral of the incident flux [**?**]

$$ka = \frac{1}{\rho_b} \int_{\Gamma_a} \vec{J}_{ss} \cdot \hat{n} \ ds. \tag{4}$$

$$\frac{\partial \rho}{\partial t} = D(\Delta \rho - \nabla \cdot (\beta \rho \ \vec{E})) \tag{5}$$

3 Mesh generation

In the first step we create a volumetric mesh based on the molecular surface of a protein. We present two options for this 1) molecularmesh (preferred) Section 3.1 2) blamer Section 3.2.

3.1 Using MolecularMesh

Firstly, creating a mesh with a globular protein and simple active site may be done using MolecularMesh (might remove this section completely once other script is developed more). Here we have provided an example using Troponin C, where we assume that the active site for calcium binding is around Glu76. For this tutorial, I will use the Troponin C structure (PDB code 1SPY). The output will be a tetrahedral mesh with important boundary conditions marked for the PDE.

PQR file. PQR files are a file format that supplements a traditional PDB structure file with atomic radii and charges, which are needed for electrostatic calculations. PQR files should be generated from the PDB files. See PDB2PQR from the APBS website or do the conversion using http://kryptonite.nbcr.net/pdb2pqr/ Here we assume p.pqr was created for TnC (PDB Code 1SPY) based on a PDB structure

Finding the active site. We first locate the position of one residues involved in binding Ca2+ (Glu76) by using 'grep':

Find active site in molecule

```
grep " 76 " tnc.pqr | grep " OE1 "
```

Store coordinates into a file called ACTIVE.DAT (replace 'coord' columns, using only 1 entry per row) and adjust 'radius' to cover your expected binding region. All other fields can be left as is.

ACTIVE.DAT

```
1
xcoord ycoord zcoord radius 1
```

Run) Now we are ready to generate the .m file, which is a volumetric mesh. MolecularMesh generates a this mesh, and is executed via:

```
MolecularMesh tnc.pqr 3 active.dat
```

This will generate a .m file named after the input tnc.pqr file.

(Proceed to Section 5 to run smol without electrostatics or Section 4 if you want to use the electrostatic potential in the association rate constant calculation)

NOTE: MolecularMesh might be broken on rocce. Please refer to Section 8.5 for a work around.

3.2 Using Blamer (Optional)

Blamer provides a more flexible framework for creating and refining meshes.

To run a blamer session

```
blender $MYDIR/gamer/tools/blender/blamer.blend
$
```

Optionally you may also create the following blamer to simplify calling blamer

```
alias blamer='export MYDIR=~/localTemp/blamer/; source $MYDIR/config.bash; blender $MYDIR/bl
```

3.2.1 Making meshes

In the following example, we show how a user would create a simulation domain for two absorbing spheres surrounded by a larger sphere representing the outer boundary. A basic knowledge of Blender is required for this section, including the creation and manipulation of primitives, selection of vertices, toggling between object and edit mode. These topics are widely addressed online.

In script window: `Scripts->Mesh->Gamer Mesh Improvements (upy)`

Surface mesh import: Create 3 spheres (non-overlapping)

Make one sphere contain the other two by changing the scale (200x200x200) NOTE: For each object, need to do `Object->Clear/Apply->Apply Scale` to make changes

Boundary tab:

- select sphere, click New, rename Name and assign Marker, select sphere, click Assign
- For outer sphere, use Marker '5' for outer boundary
- For left sphere, use Marker '1' for active site
- For right sphere, use Marker '4' for molec boundary

Tetrahedralize Tab:

- Select each mesh individuallly and click "Add selected"
- For the interior 'holes' use the "Use domain as a whole"
- Click tetrehedralize button

4 Electrostatic potential

The electrostatic potential contributes to the second term of the electro-diffusion equation. Here we describe our procedure for generating the electrostatic potential for our molecular using APBS (in the form of .dx files). While in practice any PB solver may be used, currently our interpolator script only handles dx-format APBS output.

NOTE: always be especially careful that your mesh and pdb file for the APBS calculation are well-aligned. Otherwise this creates hassles down the road.

The electrostatic potential is expected to vary quite rapidly near the molecular boundary of polar molecules and less so far from the molecular,especially in non-dilute solvents. For numerical PDE solutions, we thus require considerably higher resolution of the electrostatic field near molecule (approx 0.3 AA resolution). Meanwhile, the outerboundary should be sufficiently far from the molecular such that the electrostatic potential is neglible (about 500 Å away in some cases, for non-dilute solutions). To maintain a tractible set of inputs, we compute the electrostatic potential at a range of resolutions, then interpolate these onto our finite element mesh. One recommendation is to define the cglen/fglen keywords in the APBS input file to first compute a coarse mesh, then a fine mesh centered about your active site:

```
elec
    mg-auto
    dime 65 65 65
    cglen 1650 1650 1650  <------ coarse mesh: 1650 [A] / 65  grid points ~= 25 [A] resoluti
    cgcent mol 1
    fglen 65 65 65                 <------ fine mesh 65 [A]/65 grid points = 1.0 [A] resolut
    fgcent XXX YYY ZZZ      <------ active site location
    mol 1
    lpbe
......
write pot dx potential-0065
```

See APBS documentation http://www.poissonboltzmann.org/apbs/user-guide/ for details.

This excerpt from the APBS input file will generate generate a dx file at 1.0 [A] resolution. By using different fglen values, you can obtaining meshes that span the entire simulation domain. For this example, I'd recommend creating an input file where fglen equals a coarse value (say 1650) and a fine value (say 65) to generate two potential files (.dx):

provide example file somewhere with params

```
potential_0065.dx
potential_1650.dx
```

In practice, you'll want several meshes to ensure the potential is smooth. If the potential is not sufficiently smooth, the steady state solution from Smolfin will appear spectled (e.g. its completely wrong!).

(Proceed to Section 5.1.2 to utilize these data in smol)

5 Running smol

We present two ways to run smol:

1. Through the opal webservice Section 1.2 (easy)
2. Manually on rocce Section 5.1 (more work)

First load environmental variables

5.1 Manually running smolfin

5.1.1 Dolfin file creation

This step converts the data so far into the appropriate format for smolfin.

First load the appropriate environmental variables MAKE SPECIFIC TO ROCCE

```
export SMOLDIR=/home/huskeypm/localTemp/srcs/smol/
source $SMOLDIR/config.bash
```

To convert only the .m file to the appropriate format, run the following:

```
python mcsftodolfin.py tnc.m
```

where `tnc.m` is the file generated from MolecularMesh. This will create *file_mesh.xml.gz* and *file_subdomain.xml.gz* files needed for smolfin (Section 5.1.3)

5.1.2 Interpolation of electrostatic potential (optional)

```
python smoltodolfin.py -mcsf tnc.m <-csv electro.csv/$B> <-mgridloc [x,y,z]>
$
```

where optional arguments are listed with `< >`.

- `$B` - list of electrostatic potentials from APBS in order of increasing resolution:

  ```
  export B=’-p potential_1650.dx -p potential_0065.dx’
  $
  ```

  or you may do the following to list all files:

  ```
  export B=‘ls -t potential-*dx | perl -ane ’chomp $_; print "-p $_ "’‘
  ```

- mgridloc - where molecule center is found in molecular mesh. Usually [0,0,0], but sometimes molecular center may be off-center
- csv - csv of electro-static potential values (for debugging mostly)

This will create *file_mesh.xml.gz* and *file_subdomain.xml.gz* files needed for smolfin. You will also see *file_values.xml.gz*, which contain the electrostatic potential mapped to the finite element surface. It is recommended that you check that the potential mapped to your mesh using paraview (see the ’_values00000.vtu’ created in the working directory)

5.1.3 Running smol.py

The hard work is now done. Run smol:

```
python ~/sources/dolfin_smol/smol.py   tnc_mesh.xml.gz tnc_subdomains.xml.gz  >& run.out
```

to run without the electrostatic potential, or

```
python ~/sources/dolfin_smol/smol.py -root tnc >& run.out
```

to run with the electrostatic potential, where 'root' is the prefix for the gz files. You will obtain from this calculation an estimate of the kon value and the steady state distribution (in pvd/vtu format). The latter data may be viewed in paraview.

6 Analysis

Paraview can be used to analyze results. pvd/vtu files are generated from the SMOLFIN script which show the steady state distribution.

General instructions:

- Load pvd file into paraview
- take slice
- more details at www.paraview.org

7 Moving on

Now that you've made it here, you may wish to experiment with parameters, such as the active site location or ionic strength. You may even consider mutating acidic residues of TnC near the active site to determine their impact on the Ca2+distribution. To do so, I'd recommend using the VMD MutateResidue module.

8 Installation

8.1 Dolfin/Dorsal installation

Dorsal is a builder that contains dolfin and its dependendancies. It is available at https://launchpad.net/dorsal

dolfin (https://launchpad.net/dolfin), which is a finite element solver that my smol code uses. Technically the program can be installed on ubuntu, for instance, using apt-get, but I've had the best luck installing dolfin using the dorsal installer (https://launchpad.net/dorsal). Dorsal takes maybe 1-2 hrs to install (very little user interaction, only at very beginning) on my virtual machine, but it's probably a lot quicker on a 'real' machine.

8.2 smolfin

Smolfin uses dolfin to create meshes suitable for finite element solution and for solving the Smoluchowski equation. An early version of the code can be obtained from http://mccammon.ucsd.edu/~huskeypm/dolfin/dolfin_smol.tgz. I plan to release the software on bitbucket, so stay tuned.

8.3 APBS

APBS is used to estimate the electrostatic potential about the protein. Please consult http://www.poissonboltzmann.org for details.

8.4 paraview

Paraview is a recommended viewer for visualizing results from smolfin

To install.

```
sudo apt-get install paraview
```

Guides such as www.paraview.org provide detailed instructions for displaying values stored on the finite element mesh. For this example, the values stored by smolfin may represent the steady state distribution or the electrostatic potential.

8.5 Molec. mesh workaround

Contact me for IP address

```
ssh -XY guest@XXXXX -p 2223

  passwd: guest1979

export PREFIX=/home/huskeypm/localTemp/srcs/gamer-src/
export LD_LIBRARY_PATH=$PREFIX/lib:$LD_LIBRARY_PATH
export FETK_INCLUDE=$PREFIX/include
export FETK_LIBRARY=$PREFIX/lib
/home/huskeypm/localTemp/srcs/gamer-src/bin/MolecularMesh 1CID.pdb 3 1CID_active_site.dat
```