Sauté — a tool building membrane protein systems for molecular dynamics simulation

Thomas Mildorf and Dan Arlow

4/24/2012

# Introduction

Sauté is a collection of scripts for preparing solvated membrane protein systems for all-atom molecular dynamics simulations. Much of the functionality of Sauté, e.g. for tiling, merging, splitting mae files is available programmatically from Python via sautelib.py.

Saute is available in perforce at

//depot/desrad/user/desrad\_arlowd/saute/MAIN/src/...

or in my personal checkout at

/d/en/arlowd-0/p4workspace/user/saute/MAIN/src

# saute3.py

saute3.py is a script that takes as input a maeff file of a prepared protein structure and builds a solvated membrane system suitable for molecular dynamics simulations.

Here is the complete set of command-line options for saute3.py and default values as applicable:

Usage: saute3.py solute.maeff output.maeff

Embed solute.maeff in a tiled membrane and prepare solvent ions.

Options:

--version show program's version number and exit

-h, --help show this help message and exit

-B SOLUTE\_BB\_SEL, --solute-selection=SOLUTE\_BB\_SEL

solute.maeff atomsel to compute bounding box

[default: "ctnumber 1"]

-L LIPID\_SEL, --lipid-selection=LIPID\_SEL

atomsel for the lipids in the membrane [default:

"lipid or resname POPS"]

-C CLASH\_LIPIDS, --lipic-clash-check=CLASH\_LIPIDS

atomsel for lipids with rings (i.e. cholesterol)

that might clash with other lipids.

-M MEMBRANE\_SYSTEM, --custom-membrane-system=MEMBRANE\_SYSTEM

custom membrane system path (must be a mae file)

[default: built-in POPC + TIP3P]

-c CATION, --cation=CATION

specify cation "Na" or "K"

[default: "Na"]

--move-solute=Z\_MOVE value added to solute z coordinates

[default: 0]

-z Z\_BUF, --z-buffer-dist=Z\_BUF

buffer distance in the membrane normal direction.

[default 20.0 angstroms]

-m XY\_BUF, --membrane-buffer-dist=XY\_BUF

buffer distance through the membrane.

[default: 45.0 angstroms]

-s SALT\_CONC, --salt-concentration=SALT\_CONC

desired salt concentration.

[default: 0.150 M]

-q, --quiet

-d LIPID\_DIST, --lipid-dist=LIPID\_DIST

minimum distance from solute to lipid acyl group

[default: 1.75]

-a USER\_DIMS, --absolute-dim=USER\_DIMS

comma separated list of dimensions for system

(x and y dimensions are one number)

[default: system defined by buffers]

--absolute-xy=USER\_XY

Specifies the xy dimension. Takes precedence over

buffer-based calculation.

--absolute-z=USER\_Z Specifies the z dimension. Takes precedence over

buffer-based calculation.

-f LIPID\_FRIENDLY\_SEL, --lipid-friendly-sel=LIPID\_FRIENDLY\_SEL

atomsel for parts of the protein that are

"lipid-friendly" and should not be used when

calculating which lipids are clashing with

the protein (i.e.: lipid tails, sidechains of

peripheral membrane proteins)

saute3.py assumes that the input protein structure, solute.maeff, is positioned such that the center of the membrane is located at (0,0,0) and the extracellular surface is in the +Z direction. To build a solvated membrane system around the protein, Sauté first determines the size of the system necessary to provide the desired separation between periodic images of the protein. Since correlation lengths are different through the bilayer and water, sauté allows specification of separate “buffer” distances in the XY and Z directions. By default, sauté makes the system large enough so that there is 20 Å of water (-z/--z-buffer-dist [20 A]) between the closest atoms of periodic images of the protein in all directions. Furthermore, by default sauté makes the lateral dimensions of the system large enough so that there is at least 45 Å (-m/--membrane-buffer-dist [45 A]) between the closest atoms of the transmembrane region (abs(z) < 15 Å) of periodic images of the protein in the XY direction under all rotations of the protein about its Z axis, because the protein will naturally rotate during simulation. The user has the option to override these default parameters with desired absolute XY- and Z-dimensions (--absolute-xy and --absolute-z.)

Next, Sauté builds a solvated membrane system at least as large as the desired dimensions by tiling an equilibrated hydrated membrane patch (-M/--custom-membrane-system [default: 100% POPC; POPC+POPS with 30% POPS on -Z leaflet is also available]) in the XY plane and then deleting molecules that are outside of the desired box. The tiled membrane system is concatenated with the protein maeff (solute.maeff) and lipid and water molecules clashing with the solute (heavy atoms within 1.75 A) are deleted; optionally, a special atomsel (-f/--lipid-friendly-sel) specifies part of the solute which is ignored when determining steric clashes with lipids, (typically a lipid anchor). Special checking is performed for cyclic groups in either the solute (HIS/PHE/TRP/TYR) or the membrane (-C/--lipid-clash-check) to ensure that the system does not contain lipids that pass through rings.

Sauté adds ions by mutating water oxygens to sodium, potassium, or chloride, and deleting the hydrogens. (Waters are selected uniformly at random for mutation.) First, Sauté neutralizes the net charge of the system by adding sodium/potassium cations or chloride anions. Second, additional salt is added to achieve the requested concentration (-s/--salt-concentration [default: 0.150 M]). Either Na or K cations can be added (-c/--cation [default: Na]).

Note: in order for sauté to neutralize the system, it must compute the net charge of the solute. This is done using sum(atomsel().get(“charge”)), so the solute.maeff must have a valid force field, or sauté might produce a system with unbalanced net charge.

Finally, sauté writes the combined system output.maeff to disk.

# bias\_bond.py

bias\_bond.py is a script for adding additional harmonic (i.e. “bond”) terms to a maeff file.

Here is the complete set of command-line options for bias\_bond.py:

Usage: bias\_bond.py in.maeff out.maeff bias.cfg

bias\_bond.py in.maeff out.maeff selection x0 k

Options:

--version show program's version number and exit

-h, --help show this help message and exit

-b C\_T, --ct-block=C\_T

c\_t block to restrain (default: 1)

There are two ways to specify harmonic terms to bias\_bond.py: you can provide an ark-formatted config file containing several selections and parameters, or you can specify the atoms and parameters for a single term on the command line. In either case, you provide three things for a bond term: a VMD atomsel string that evaluates to precisely the two atoms you want to include in the term, an equilibrium length “x0” and a force constant “k”. In the single-term usage, you provide the three values on the command line. In the config file usage, you provide an ark with a top-level list called “bias” containing length-3 lists of the form “[selection x0 k]”. An example bias ark is:

bias = [

["protein and name CA and resid 75 323" 10.58 2.0]

["protein and name CA and resid 47 319" 5.90 4.0]

]

This ark causes bias\_bond.py to add two harmonic terms between two different pairs of Cα atoms with equilibrium lengths 10.58 Å and and 5.90 Å and force constants of 2.0 and 4.0 kcal/mol/Å2, respectively.

NOTE: the Anton parameter “anton.chem.maxBondLength” must be set to a value greater than the largest distance of atoms involved in any bond in the system in any timestep during simulation. The default value for anton.chem.maxBondLength is 4.0 Å and should not be increased much beyond ~15 Å because large values will result in decreased accuracy of the calculation of all bonded forces throughout the system. This consequently limits what length and stiffness of terms you can add with bias\_bond.py.

# bias\_dihedrals.py

bias\_dihedrals.py is a script for adding either harmonic or gaussian-like terms to dihedral angles of four atoms in a maeff file.

Here is the complete set of command-line options for bias\_dihedrals.py:

Usage: bias\_dihedrals.py bias.cfg in.maeff out.maeff

bias ark format is bias = [[dihedral\_atomsel phase depth] ...]

Options:

--version show program's version number and exit

-h, --help show this help message and exit

-b C\_T, --ct-block=C\_T

c\_t block to restrain (default: 1)

-r, --harm-restrain restrain dihedrals harmonically

(default: use gaussian-like potential)

Bias terms to be added are specified in an ark-formatted config file as a list length-3 lists of the form “[selection phase k]”. “selection” is a VMD atomsel string that evaluates to precisely the four atoms you want to include in the term, “phase” θ0 is the equilibrium angle, and “k” is a well depth (in kcal/mol) or a force constant (in kcal/mol/degree2) for gaussian-like terms and harmonic terms, respectively. An example bias ark is:

bias = [

["protein and resid 208 and name N CA CB CG" 180.0 5.0]

["protein and resid 289 and name N CA CB CG" 60.0 -3.0]

]

This ark causes bias\_dihedrals.py to add two terms to the χ1 dihedrals of residues 208 and 289 with equilibrium angles 180.0º and 60.0º and well depths of 5.0 kcal/mol and -3.0 kcal/mol respectively.

The gaussian-like terms added are of the form:

And the harmonic terms added are of the form: