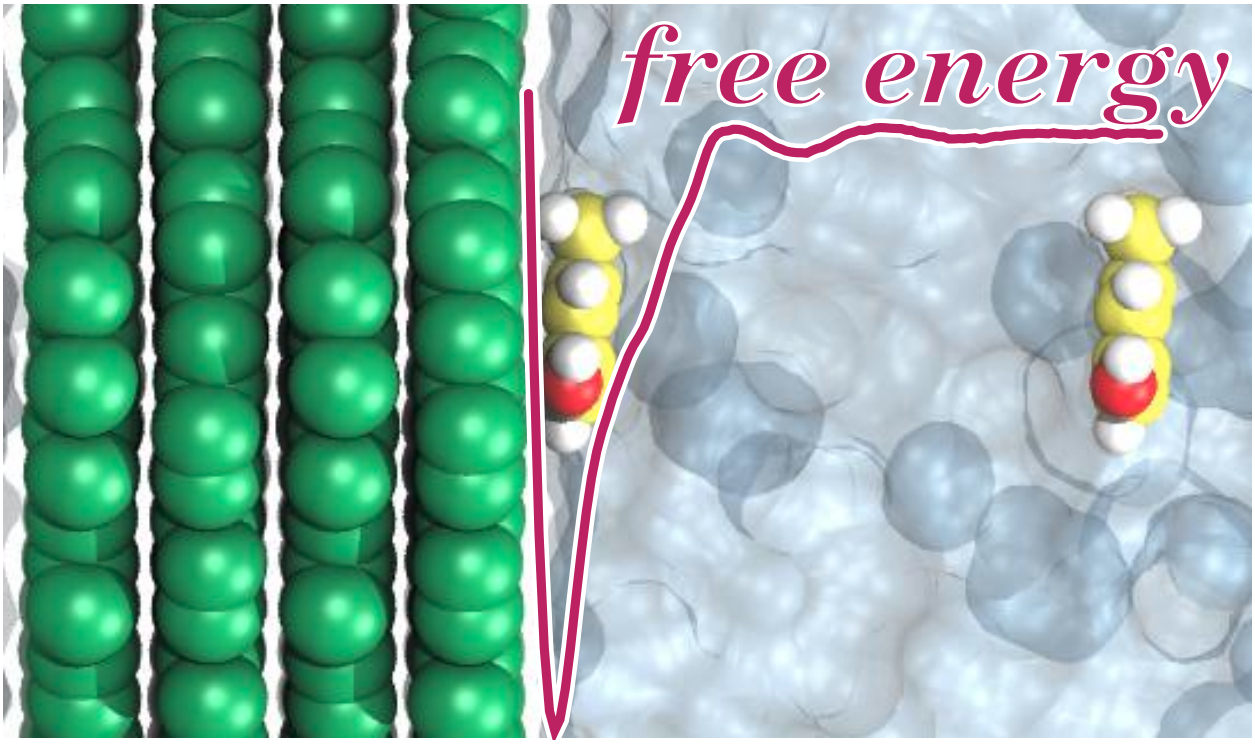
Fast Free Energy Calculations of Binding to Surfaces

A tutorial for users of VMD and NAMD.



Horacio Poblete
Jeffrey Comer
Nanotechnology Innovation Center of Kansas State
Institute of Computational Comparative Medicine
Department of Anatomy and Physiology
Kansas State University

March 27, 2018

1 Introduction

This tutorial is designed to guide users of VMD and NAMD in all the steps required to set up a free energy calculations for adsorption on graphene. The tutorial assumes that you already have some knowledge of VMD and NAMD, as well as of the basics of parameterization of organic molecules and/or proteins.

Knowing the adsorption free energy of molecules on surfaces is crucial in designing new materials able to capture/release chemical compounds in a selective way. We have recently published results [1] demonstrating that molecular dynamics simulations coupled with free-energy calculation techniques can predict adsorption affinities in good agreement with experiment. Therefore, we have designed this tutorial to show how one can calculate the free energy for adsorption small organic molecule (biphenyl), to a graphene surface. We will not detail all aspects of performing these calculations, which can require some subtle considerations. We urge readers to refer to the cited articles for more information on these methods and good practices in their use. The NAMD and VMD tutorials [`http://www.ks.uiuc.edu/Training/Tutorials/`] are also essential sources for understanding the use of these programs. The Colvars manual provides additional information on the use of the Colvars module and can be downloaded from the Colvars github page [`http://colvars.github.io/`]. In this tutorial, we will use the program VMD [2] to generate the graphene surface through the InorganicBuilder plugin [3, 4]. We will then parameterize the organic molecule in the framework of the CHARMM General Force Field (CGenFF) [5] using the ParamChem server [6, 7]. Subsequently we will use the adaptive biasing force (ABF) methodology [8, 9] to calculate the potential of mean force (PMF) and estimate the free-energy of binding of biphenyl on the surface of graphene. Note that the ABF methodology can be used in an endless number of situations, so this tutorial will give you the basic tools to perform a variety of free energy calculations.

ABF calculations are reliant on the definition of one or more transition coordinates, along which sampling is enhanced. In this tutorial, we use the Collective Variable-based Calculations Module (Colvars) [10] to define a transition coordinate that is the distance between the center of mass of an organic molecule and a graphene surface. The Colvars module can be used with NAMD (as well as with VMD and LAMMPS) to reduce the large number of degrees of freedom of a physical system into few parameter whose statistical distributions can be analyzed individually. Furthermore, it permits the application of biases to alter the dynamics of the system in a controlled manner, such as those that are used in the free-energy calculation methods of ABF [9], metadynamics [11], and umbrella sampling [12, 13]. The various free-energy calculation methods have different advantages and disadvantages. For the problem at hand, we have chosen ABF, which, distinct from the other methods mentioned, relies on estimating forces on the transition coordinate, providing a convenient physical framework and a rigorous guarantee of long-time convergence [14, 15]. We these tools, we estimate the PMF and consequently the free-energy of binding for a small organic molecule in a graphene surface.

Getting Started

This tutorial requires the visualization software VMD [2] and molecular dynamics program NAMD [16], which can be downloaded for free to run on most platforms.

Completion of this tutorial requires:

- Various files contained in the archive `TutorialFreeEnergy_files.zip`.
- VMD: `http://www.ks.uiuc.edu/Research/vmd/`

- NAMD: `http://www.ks.uiuc.edu/Research/namd/`
- GRACE: For making plots, the free program `xmgrace` (Linux) is recommended, it can be downloaded from: `http://plasma-gate.weizmann.ac.il/Grace/`

2 Building the system

In this unit, you will construct a model of a graphene surface covered in a layer of water and add a small organic molecule to the system.

2.1 Constructing multi-layer graphene.

In this section, we will use the InorganicBuilder plugin of NAMD to construct a small patch of multi-layer graphene.

1. Open InorganicBuilder by selecting Extensions → Modeling → InorganicBuilder from the VMD menu.
2. Select Task → Build device, to begin building.
3. Choose the material Graphite and check Hex box.
4. Enter the Hex box dimensions D: 12 and Z: 2.
5. The InorganicBuilder window should look like that below. When you are ready, click the Build device button.

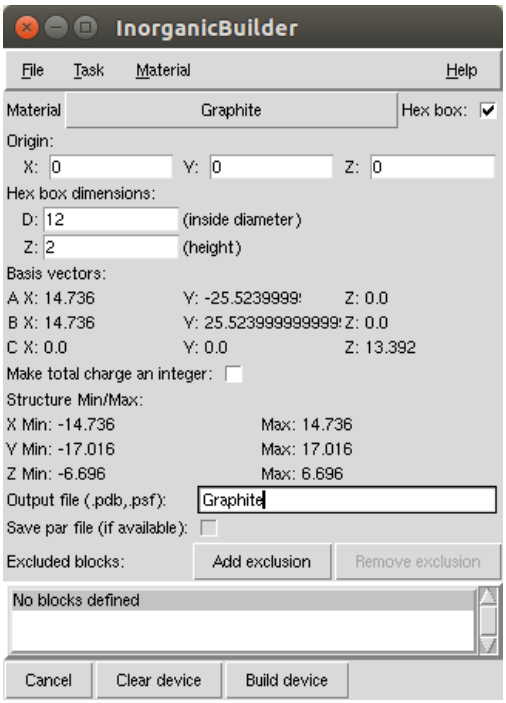


Fig. 1: Example of using InorganicBuilder to build a multi-layer graphene system.

2.2 Adding bonds to graphene

Here we continue with the InorganicBuilder plugin to add bonds to the graphene structure.

1. Select Task → Add bonds from the InorganicBuilder menu.
2. Select Ignore existing bonds, build only specified bonds.
3. Click the Select loaded molecule button and choose `Graphite.psf`.
4. The basis vectors should already be correct.
5. The bonds should be generated periodic in A and B, but not in C (along the z axis). The top and bottom surfaces (perpendicular to the z axis) will form the interface with water. Deselect Periodic in C.
6. Select Transform to hex, Build angles, Dihedrals.
7. Set Output file to Graphite_bonds.
8. Click the Add bond button and add a bond between Atom 1: C and Atom 2: C with a maximum bond length of 1.45 Å (the distance between carbon atoms in graphene is 1.418 Å. After setting Bond length to 1.45 Å. Click Add.
9. The InorganicBuilder window should look like that below. When you are ready, click the Find Bonds button.

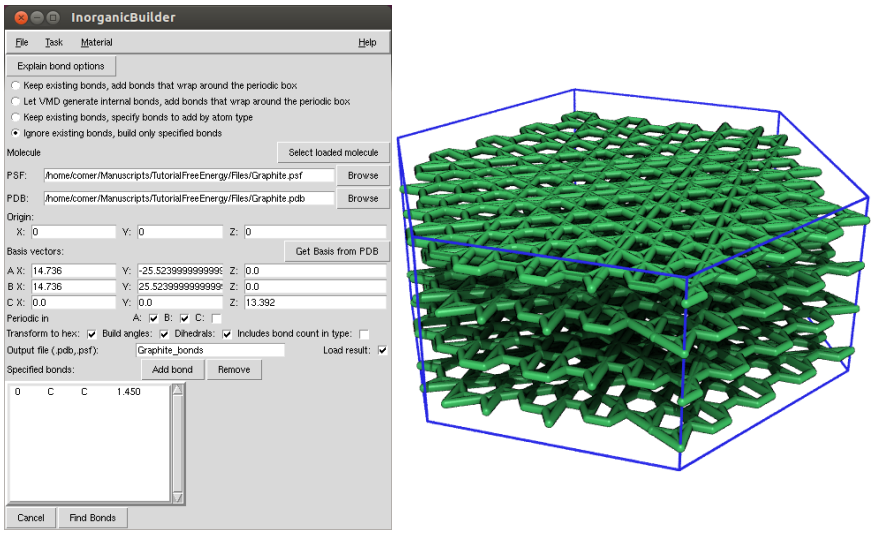


Fig. 2: Example of using InorganicBuilder to add bonds to the graphene system.

10. You should now see bonds between carbon atoms, including some that appear to cross the entire system as above. While the latter bonds may look strange, they are a result of the periodic boundary conditions—the bonds don't actually extend across the whole system, but are between atoms on one apparent edge of the hexagon and another apparent edge. With periodic boundary conditions, there are truly no edges—the graphene surface is *effectively infinite*, covering the entire xy plane. As shown below, the periodic boundary conditions can be visualized by selecting Graphics → Representations..., selecting the Periodic tab, and adding periodic images at +X and +Y.

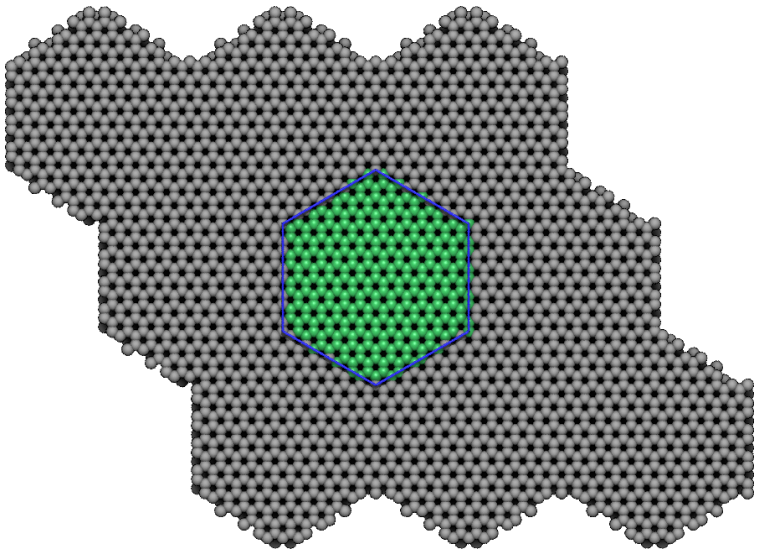


Fig. 3: Periodic images of the hexagonal graphene system. The periodic boundary conditions yield an effectively infinite graphene surface.

11. We can check that the bonds have been correctly assigned by making sure all carbon atoms are bonded to 3 others, consistent with the topology of graphene. Select Extensions → TkConsole and enter the following:

```
set s [atomselect top "numbonds != 3"]
puts "Bad atoms: [$s num]"
```

The number of atoms having less or more than three bonds should be zero. If not, there is a problem with the assignment of bonds in the last step, perhaps having to do with the system basis vectors.

2.3 Preparing the multi-layer graphene–water system.

Here we solvate the graphene surface using the VMD Solvate plugin.

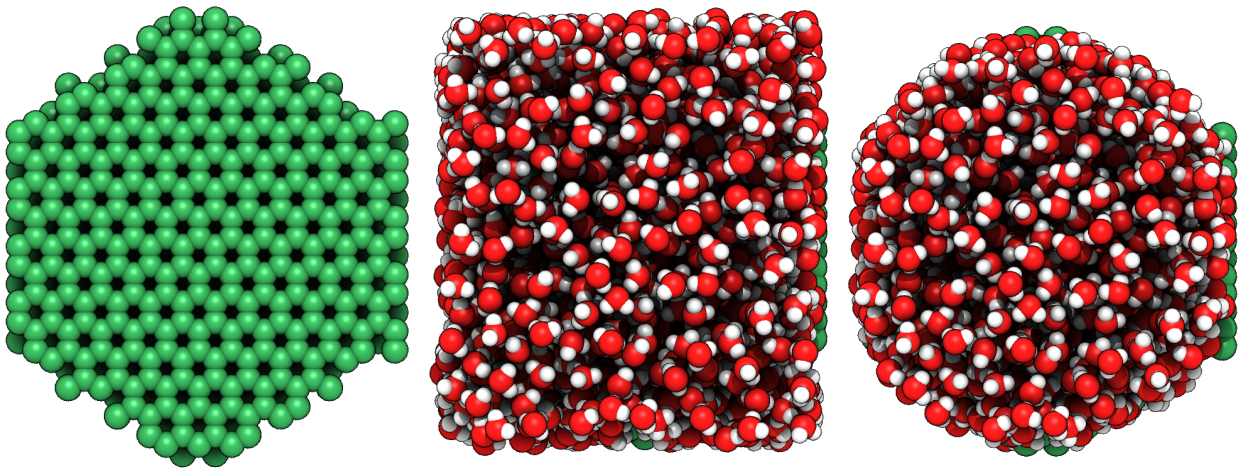


Fig. 4: Solvating the multi-layer graphene system.

1. Enter the following in the TkConsole to move the center of mass of the multi-layer graphene to the position $(0, 0, -11)$, which we do for convenience.

   ```
   set all [atomselect top all]
   $all moveby {0.0 0.0 -11.0}
   ```

2. We need to define the "type" of the graphene atoms, which determines how they interact with each other and other atoms in the system. We represent graphene atoms with the CHARMM General Force Field (CGenFF) [5] type for 6-member aromatic rings: CG2R61. To set this parameter type for our multi-layer graphene, enter the following in the TkConsole:

   ```
   $all set type CG2R61
   ```

3. Write the modified structure.

   ```
   $all writepsf Graphite_ready.psf
   $all writepdb Graphite_ready.pdb
   ```

4. Now we'll add water molecules to the system using the Solvate plugin of VMD. Enter the following in the TkConsole to build a box of water around the system.

   ```
   package require solvate
   set box {{-15 -17 -23} {15 17 23}}
   solvate Graphite_ready.psf Graphite_ready.pdb -minmax $box -o Graphite_sol
   ```

5. The box of water that we've created clearly does not conform to the hexagonal shape of our system. Here we cut the system to a cylindrical shape, which serves as a sufficiently accurate approximation for this small system. Energy minimization performed with NAMD should remove any clashes between water molecules, provided that they aren't too drastic.

   ```
   set s [atomselect top "resname C or (same residue as x^2+y^2<15.5^2)"]
   $s writepsf Graphite_hex.psf
   $s writepdb Graphite_hex.pdb
   ```

6. Have a look at the new system by entering the following in the TkConsole.

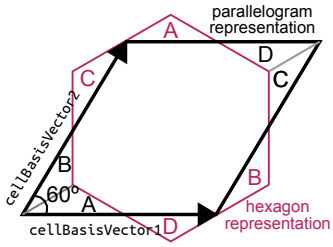
   ```
   mol delete all
   mol new Graphite_hex.psf
   mol addfile Graphite_hex.pdb
   ```

   There shouldn't be any water molecules within the multilayer graphene membrane, which may become stuck between layers of graphene during minimization. If there are, they could be removed by replacing the selection text above with something like:

   ```
   "resname C or (same residue as (x^2+y^2<15.5^2 and abs(z+11)>7))"
   ```

Basis vectors

As can be seen below, the hexagonal system that we've created has an equivalent parallelogram representation. The parallelogram representation gives the system basis vectors that we need for simulation with NAMD.



2.4 Modeling the solute molecule.

We will use the ParamChem server to generate the topology and parameter files for our organic molecule: biphenyl. Biphenyl has a quite high affinity for the graphene surface (the adsorption free energy, $\Delta G_{\text{ads}} \ll -k_{\text{B}}T$), which determines the available methods for determining the ΔG_{ads} as is discussed in more detail in the section "Performing the Free Energy Calculation."

1. First, it is necessary to have a structure file with the molecule that you want to parameterize. The mol2 format is best for this purpose because it contains information (such as the presence of double bonds) that is not available in a pdb file. Such mol2 structures can be constructed using the Molefacture plugin of VMD (Extensions → Modeling → Molefacture). Another option is to obtain the structure in mol format (by clicking the 3D button) from the ChemSpider database (`http://www.chemspider.com/`) and convert it mol2 format using software such as OpenBabel. Here we provide you an example `BPH.mol2`, shown below.

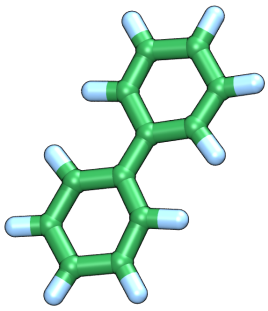


Fig. 5: Three-dimensional structure of biphenyl

2. Go to the website of ParamChem: `http://www.cgenff.paramchem.org` First of all, you should create you own account in ParamChem, only after that you will be able to do the following steps.

3. Activate you account, then login.

4. Select Browse... → select your molecule, to upload the file `BPH.mol2`.

5. Download the output file that contains topology and parameters, saving it as `cgenff_bPh.str`.

6. Open `cgenff_bPh.str` in a text editor, change the resname **/scrat by → BPH**. Below we show an abriged version of the final topology file `cgenff_bPh.str`.

```
* CHARMM General Force Field (CGenFF) program version 1.0.0
* using valence-based bond orders
*
36 1

! "penalty" is the highest penalty score of the associated parameters.
! Penalties lower than 10 indicate the analogy is fair; penalties between 10
! and 50 mean some basic validation is recommended; penalties higher than
! 50 indicate poor analogy and mandate extensive validation/optimization.

RESI BPH         0.000 ! param penalty=   0.000 ; charge penalty=   0.000
GROUP            ! CHARGE   CH_PENALTY
ATOM C1     CG2R67 -0.004 !    0.000
ATOM C2     CG2R67 -0.004 !    0.000
.
.
ATOM H9     HGR61   0.115 !    0.000
ATOM H10    HGR61   0.115 !    0.000
               ! Bond order
BOND C10  H8   ! 1
BOND C10  C12  ! 1
.
.
BOND C6   C2   ! 1

END
```

7. ParamChem assigns parameters based on analogies between the chosen compound and well parameterized model compounds in the CGenFF library. The "penalties" given above estimate the quality of these parameters [6, 7]. For some structures, the penalties may be high and additional work may be needed to improve them, which is outside the scope of this tutorial.

8. In this step, we are going to build the connectivity file (psf) for our molecule biphenyl, using the topology generated by ParamChem. For this, we need to be in the same folder the `BPH.mol2, BPH.str,top_all36_cgenff.rtf` and the script `01Build_psf_bph.tcl`.

9. Open VMD (if you have closed it). Now we will load the mol2 file and change the residue name and atom names to be consistent with the parameter file `BPH.str`. Enter the following in the TkConsole.

```
mol new BPH.mol2
```

```
set all [atomselect top all]
$all set segname A
$all set resname BPH
set nL {C1 C2 C3 C4 C5 C6 C7 C8 C9 C10 C11 C12 H1 H2 H3 H4 H5 H6 H7 H8 H9 H10}
$all set name $nL
$all writepdb BPH.pdb
```

10. Now we will use the psfgen plugin of VMD to construct the psf. The script `01Build_psf_bph.tcl` contains the following commands.

```
set out bph_psf

# Topology Files
package require psfgen
topology top_all36_cgenff.rtf
topology cgenff_bPh.str

# Use psfgen to build the segment.
segment A {first none; last none; pdb BPH.pdb; auto angles dihedrals}
coordpdb BPH.pdb A

# Write the new PDB & PSF files.
writepdb $out.pdb
writepsf $out.psf
```

Assembly of psf files.

In general this is possible for any number of psf and pdb files that you wish to merge, but is mandatory that the segments in each psf and pdb are different.

11. Open VMD → Extensions → Tk Console and write:

```
source 01Build_psf_bph.tcl
```

or directly from the Unix terminal enter:

```
vmd -dispdev text -e 01Build_psf_bph.tcl
```

12. Assemble a final psf and pdb for the complex graphene surface-biphenyl-water.

We will open the software VMD and display the Tk console, then write:

```
resetpsf
readpsf bph_psf.psf
coordpdb bph_psf.pdb
```

```
readpsf Graphite_hex.psf
coordpdb Graphite_hex.pdb

writepsf graph_bPh_wat.psf
writepdb graph_bPh_wat.pdb
```

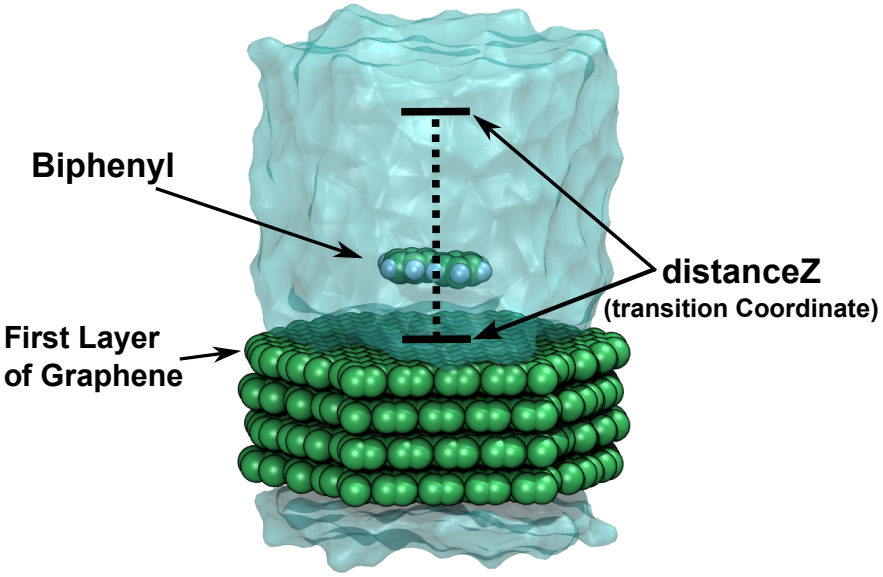


Fig. 6: Final system ready to run on NAMD.

3 Using the NAMD Colvars Module

In this unit, we'll learn how to build a Colvars configuration file to track variables in the system, apply restraints, and perform free-energy calculations.

The Colvars configuration file contains everything that the In this section, we will generate a selection of atoms involved in our collective variable. So, we will write a file containing the instructions to be executed during the molecular simulation run. In addition we will include definitions for ABF.

3.1 The Colvars configuration file

1. To obtain the serial numbers of the atoms used to define the collective variable, type the following into the VMD TkCon:

```
mol new graph_bPh_wat.psf
mol addfile graph_bPh_wat.pdb
set sel1 [atomselect top "resname BPH"]
$sel1 get serial
set sel2 [atomselect top "resname C and z>-7"]
$sel2 get serial
```

You should be able to see two lists for sel1 (1 to 22) that corresponds to the serial numbers of all atoms of biphenyl and for sel2 (28 to 1174) that corresponds to the atoms in the top layer of graphene.

2. Now we will write the colvars file, which is an external file that is read by NAMD.

```
# File: abf_3_15_bPh.colvars
colvarsTrajFrequency     500
colvarsRestartFrequency  20000

colvar {
   name NormalMolMem

   width 0.05

   lowerBoundary 3
   lowerWallConstant 20.0
   upperBoundary 15
   upperWallConstant 20.0

   outputValue
   outputAppliedForce

   distanceZ {
      main {
atomNumbers { 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 }
      }
      ref {
atomNumbers { 28 30 36 38 44 46 52 54 60 62 68 70 76 78 84 86 92 94 100 102 108
110 116 118 124 126 132 134 140 142 148 150 156 158 164 166 172 174 180 182 188
190 196 198 204 206 212 214 220 222 228 230 236 238 244 246 252 254 260 262 268
270 276 278 284 286 292 294 300 302 308 310 316 318 324 326 332 334 340 342 348
350 356 358 364 366 372 374 380 382 388 390 396 398 404 406 412 414 420 422 428
430 436 438 444 446 452 454 460 462 468 470 476 478 484 486 492 494 500 502 508
510 516 518 524 526 532 534 540 542 548 550 556 558 564 566 572 574 580 582 588
590 596 598 604 606 612 614 620 622 628 630 636 638 644 646 652 654 660 662 668
670 676 678 684 686 692 694 700 702 708 710 716 718 724 726 732 734 740 742 748
750 756 758 764 766 772 774 780 782 788 790 796 798 804 806 812 814 820 822 828
830 836 838 844 846 852 854 860 862 868 870 876 878 884 886 892 894 900 902 908
910 916 918 924 926 932 934 940 942 948 950 956 958 964 966 972 974 980 982 988
990 996 998 1004 1006 1012 1014 1020 1022 1028 1030 1036 1038 1044 1046 1052 1054
1060 1062 1068 1070 1076 1078 1084 1086 1092 1094 1100 1102 1108 1110 1116 1118
1124 1126 1132 1134 1140 1142 1148 1150 1156 1158 1164 1166 1172 1174 }
      }
   }
}
```

Let's go through the options one by one. First, `colvarsTrajFrequency` is the number of steps between output to the `.colvars.traj` file. This file records the step number, value of the

collective variable, and the value of biasing forces. `colvarsRestartFrequency` is the number of steps between output to the `.colvars.state` file, which allows the biasing scheme to be restarted if the simulation is stopped prematurely. Below this, we define a collective variable with the keyword `colvar` called `NormalMolMem`. The keyword with defines the size of the bins used to collect the system force on the collective variable as prescribed by the ABF method. Here we use 0.05 Å, which is able to capture the small details in the free-energy profile. `lowerBoundary` and `upperBoundary` define the region of the collective variable over which ABF is applied. Here, we want to obtain the free-energy profile from a distance of 3 Åfrom the surface to a distance of 15 Å. Defining `lowerWallConstant` and `upperWallConstant` mean that a force is applied to keep the collective variable within the region of interest. In this case, we apply a half harmonic force with a spring constant of (20 kcal/mol)/(0.05 $Å^2$). `outputValue` and `outputAppliedForce` indicate that the value of the collective variable and the biasing force, respectively, are written to the `.colvars.traj` file. The type of collective variable we are defining is a `distanceZ` variable, which gives the component of a vector between two points projected along an axis (by default, this is the z axis (0,0,1)). `main` and `ref` define the points from which this vector is calculated. In our case, these two points are the center of mass of the biphenyl molecule and the center of mass of the first layer of graphene. Please see the Colvars manual for more information [`http://colvars.github.io/`].

Atom numbering

In colvars, as in NAMD, atoms numbering begins from 1, whereas in VMD, it begins from 0 (index number). An easy way to get the 1-based numbers in VMD is to use the keyword serial instead of index.

3. To turn on the ABF method, in the same colvars file, we add the following lines:

```
abf {
   colvars NormalMolMem
   fullSamples 500
   historyFreq 20000
}
```

In simple terms, with these lines we allow to ABF to read and record samples of the force along the transition coordinate (collective variable). The mean force can be calculated as a function of the collective variable by averaging over the force samples in bins of width 0.05 Å. The opposite of this estimated mean force is applied to the system, flattening free energy barriers and enhancing sampling. As the estimated mean force approaches the true mean force, the mean force can be integrated to obtain the free-energy profile, also known as the potential of mean force (PMF). `fullSamples` determines how many samples much be accrued in each bin before the bias force is applied. `historyFreq` determines how often the current estimate of mean force, number of samples, and PMF are written to the `.hist.grad`, `.hist.count`, and `.hist.pmf` files.

4 Performing the Free Energy Calculation

In principle, the free energy of adsorption of biphenyl can be calculated by running a long equilibrium simulation and comparing the probabilities of finding biphenyl near the surface (e.g., $3.4 < z < 3.6$ Å) and far from the surface (e.g., $14.8 < z < 15.0$ Å):

$$\Delta G = -k_{\mathrm{B}} T \ln \frac{P_{\mathrm{near}}}{P_{\mathrm{far}}}. \tag{1}$$

This ΔG can then be used to predict surface densities of biphenyl for given aqueous concentrations or compare affinities between different molecules and surfaces. However, the adsorption affinity is sufficiently high that, in an equilibrium simulation, the molecule would simply adsorb to the surface and stay there for the rest of the simulation, precluding our ability to estimate $P_{\mathrm{near}}/P_{\mathrm{far}}$. In fact, as detailed in the Supporting Information of Comer et al. [1], we calculate that the escape of biphenyl from the graphene surface should typically require ∼10 μs. Thus, properly sampling $P_{\mathrm{near}}/P_{\mathrm{far}}$ would require simulations of ∼100 μs, much longer than those available to most researchers. Equilibrium simulation is therefore not well suited to this problem. On the other hand, ABF and other free-energy calculation methods invoke "importance sampling": the simulation is performed on a biased free-energy landscape, designed to yield uniform sampling over the coordinate of interest. Hence, ABF will allow us to obtain an accurate prediction of the free energy for biphenyl adsorption.

1. **Generation of a restraint file for the graphene surface** In this step we will generate a restraint file for the two intermediate layers of the graphene surface. Applying these restraints is not entirely necessary; however, without them the graphene would drift along the z axis. Because the collective variable is defined in terms of distances between the first layer of graphene and the biphenyl molecule, this drift should not affect the free-energy calculation. However, it can make visualization and analysis somewhat more difficult. Note that we do not restrain the top layer of graphene, which could artificially suppress its deformation, modifying slightly its interaction with biphenyl. (As shown in the Supporting Information of Comer et al. [1], such restraints may have very small but, perhaps measureable, effects on the PMF.) To apply these restraints, we need to first select in VMD the atoms that we want to restrain.

   (a) Open VMD and load psf and pdb files (`graph_bPh_wat.psf` and `graph_bPh_wat.pdb`).

   (b) From the VMD menu, choose Graphics → Representations...

   (c) Create a representation `resname C and z<-8 and z>-15`, to visualize the two intermediate layers of graphene.

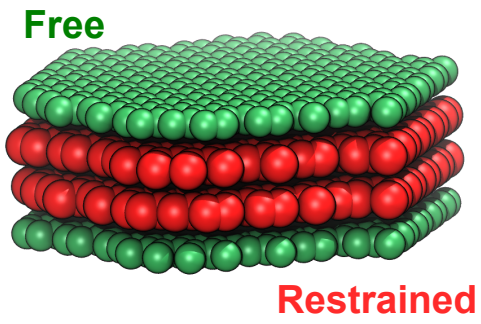


Fig. 7: Restraint to apply during the ABF simulation.

(d) Now that we are sure about our selection go to: Extensions → Tk Console and enter: `source 07Restraint_CA.tcl`
This script will generate a pdb with the atoms that we visualized above marked in their beta columns, allowing NAMD to apply a 1 kcal/mol restraint to each atom selected.

```
set all [atomselect top all]
$all set beta 0
set sel [atomselect top "resname C and z<-8 and z>-15"]
$sel set beta 1
$all writepdb rest_bPh.pdb
exit
```

2. **Execute the ABF calculations** Now that our structure (psf) file is ready and that a transition coordinate has been defined in the Colvars configuration file, and we have prepared the restraint pdb, we will execute out calculations with NAMD. In the folder

   `TutorialFreeEnergy_files/03_run/namd/`

   you will find three files ready to run: 10000 steps of minimization, 200 picoseconds of equilibration and 100 nanoseconds of ABF. The first two steps can be executed on a single processor (about 20–60 minutes, each), but the last step of ABF may take several days on a single processor computer. If this is too long, an example of what would be obtained from running them to completion is available in the `example/graph_bPh_wat.2.dcd` folder.

Understanding the NAMD configuration files

Please check each NAMD file before run the molecular simulation. The simulation is performed with a Langevin thermostat [16] to maintain the temperature at 300 K. The pressure is maintained at 1 atm using the Langevin piston method [17]. Because of the graphene covers the entire xy plane, the fluctuations of the system size associated with pressure control apply only along the z axis. Full electrostatic interactions are computed via the particle-mesh Ewald algorithm, [18] with a mesh spacing of < 1.2 Å. We use a smooth cutoff from 8 to 9 Å of the van der Waals forces and the direct portion of the electrostatics computation. The use of this cutoff scheme is validated in the Supporting Information of Comer et al. [1].

All the files *.namd will be executed consecutively by commands like the following:

```
namd2 input.namd > output.log &
```

(a) `min_3_15_graph_bPh.0.namd` → 1000 steps

(b) `equil_3_15_graph_bPh.0.namd` → 200 ps equilibration

(c) `abf_3_15_graph_bPh.1.namd` → 100 ns ABF

Each NAMD run will produce files necessary for the following run.

It is possible to plot the `*.count` and `*.pmf` file that is produced during the molecular simulation, in which you will be able to see the progress in the sampling and in the determination of the PMF, respectively. Also, you can plot the `*.hist.pmf` and check how the PMF changes as the calculation proceeds.

5 Validating the Free Energy Calculation

There are many problems that can occur in free-energy calculations that can make them yield erroneous results. The results are always dependent on the accuracy of the force field describing the interactions between atoms, and this can be difficult to control without huge effort (force field development). The quality of the sampling is often the most important issue that concerns the practitioner. While it is difficult to eliminate all possible sources of error, good practice suggests that we look for all obvious signs of poor sampling that may invalidate our results. Below we check whether the full interval of the transition coordinate is well sampled, whether the biphenyl molecule obtains adequate conformational sampling, and whether the ABF method appears to converge in time and between two independent simulations. Signs of inadequate sampling or poor convergence are red flags indicating that we need to perform longer simulations or, worse, that there is some flaw in the design or protocol of our simulation.

1. **Sampling along the transition coordinate.** Plot the count file produced by Colvars by typing the following command in the Unix terminal.

   ```
   xmgrace abf_3_15_graph_bPh.2.count &
   ```

   The sampling should be roughly uniform. A rule of thumb is that no region should receive less than half of the mean level of sampling. You should see that regions should have more than about 5×10^5 samples. If you have any regions with only a few thousand samples, the mean force will not be accurate, and differences in free energies over intervals including this region will not be reliable. The uniformity of sampling can be improved by running longer or by the use of windows [9] (running distinct ABF calculations over several smaller intervals and combining the results).

2. **Observing transitions along the transition coordinate** Another simple indication of good sampling is observation of the dynamics of the transition coordinate. While viewing the count file above can show that the system spent about the same time in all regions of the transition coordinate, it does not indicate how often transitions between these regions occurred. Plot the trajectory of the transition coordinate.

   ```
   xmgrace abf_3_15_graph_bPh.2.colvars.traj &
   ```

   By plotting the `.colvars.traj` file we can see that the biphenyl molecule traveled between the surface $z < 3.5$ Å and the bulk aqueous solution $z > 14$ Å many times.

3. **Conformational Sampling.** So now we are fairly sure that the transition coordinate has been well sampled. However, there are other coordinates orthogonal to the transition coordinate that also must be well sampled for the calculated free energy to be reliable. For example, we need to sample all accessible orientational states of biphenyl. The easiest way to check conformational sampling is to display all conformations that the biphenyl molecule reached during the trajectory. For this we will open VMD and display the following:

   (a) Load on VMD the psf file

   `03_run/graph_bPh_wat.psf`

   and the dcd file

   `03_run/output/graph_bPh_wat.2.dcd}`

   (b) Open Graphical Representations by selecting VMD → Graphics → Graphics Representations

   (c) Create two representations:

      i. resname C

      ii. noh resname BPH

   (d) Go to: Graphics Representation → Trajectory and replace now by 0:2:2000
   This means that VMD will show you all the frames from 0 to 2000 every 2 frames. The Graphical Representations window should look like that below.

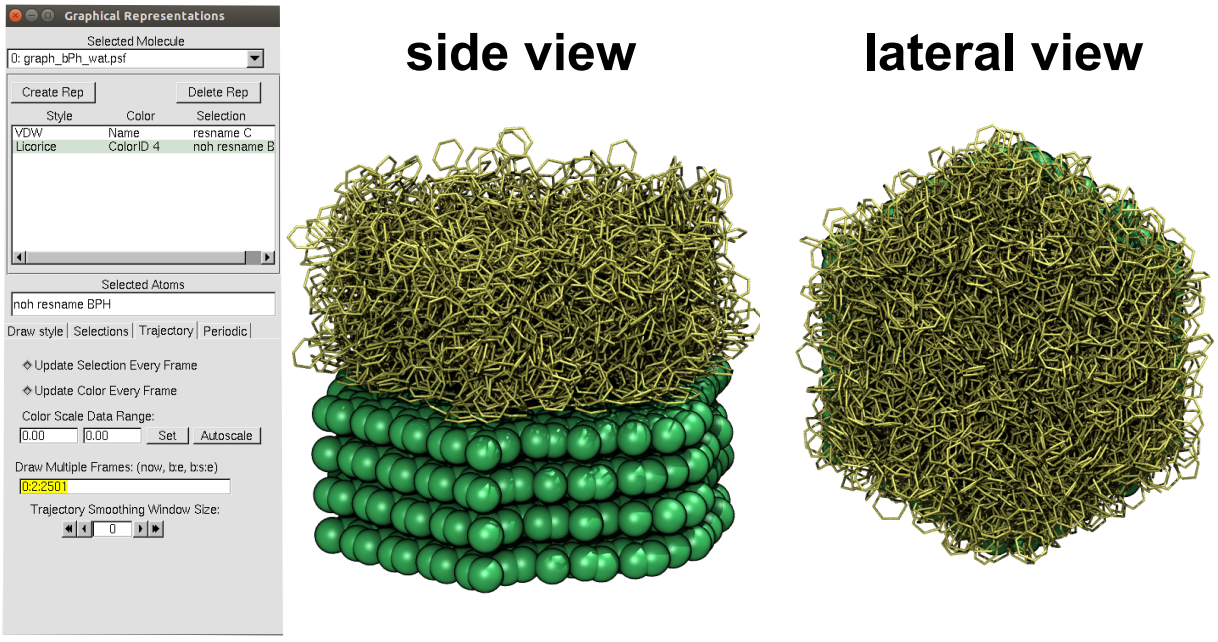


Fig. 8: Conformational sampling of biphenyl on a graphene surface.

4. **Checking convergence in two sections of the same trajectory.**

   For this purpose, we will use the `abfCheck.tcl` script, which basically counts the number of data stored in the historical files (grad and count), and extracts the data for half and

the end of the trajectory. Then the value of the gradient obtained, the script is able of calculating the value of the PMF in both windows. Besides the script normalizes PMF to zero value assuming a energetic plateau at a distance more than 12 Åwhere is no an interaction between the biphenyl and graphene surface.

To execute the script you can write in the terminal:

```
./abfCheck.tcl 11.5 12.0 abf_3_15_graph_bPh.2.hist.grad out_check_pmf
```

Now you can plot the two pmf files generated and check the superimposition of the curves.

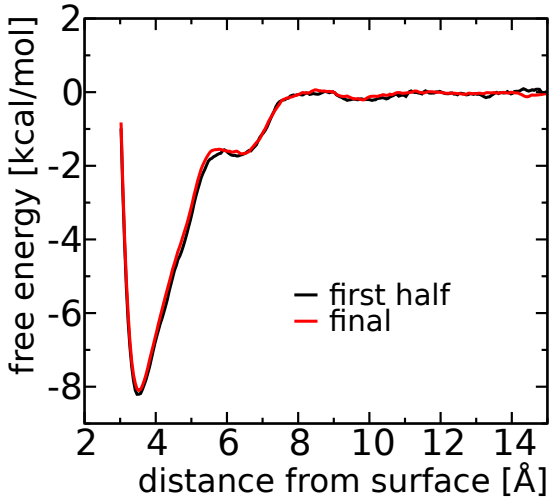


Fig. 9: Comparison of convergence one a single trajectory.

5. **Comparing two independent calculations.**

   Like we said before, comparing two portions of the same trajectory is a good parameter for estimate convergence. On the other hand, a good practice to check rigorously the convergence is to make replicas of the ABF calculation starting from different initial conformations.

   For this tutorial, we have generated two trajectories started from different initial conformations. The second trajectory you can find it in the folder: 05_replica. With this we will plot the final pmf files of each trajectory and check the overlap of the curves.

   (a) In this case, we can use the script `getColvarHistoryFrame.tcl` to extract from the historical file the pmf at different frames of each simulation and compare the overlaping of the curves.

   ```
   tclsh getColvarHistoryFrame.tcl abf_3_15_graph_bPh.2.hist.pmf 50
   hist_frame/pmf_50
   ```

   As example we extract the frames for 50, 200, 500, 1000 and 2000, which are shown in the Figure 10. Also, we plot the last two PMF of each simulation (bottom, Fig.

10), noting that both curves have an acceptable overlap. This plot gives us an idea of the uncertainty (statistical error) in our calculations, which appears to be roughly 0.3 kcal/mol.

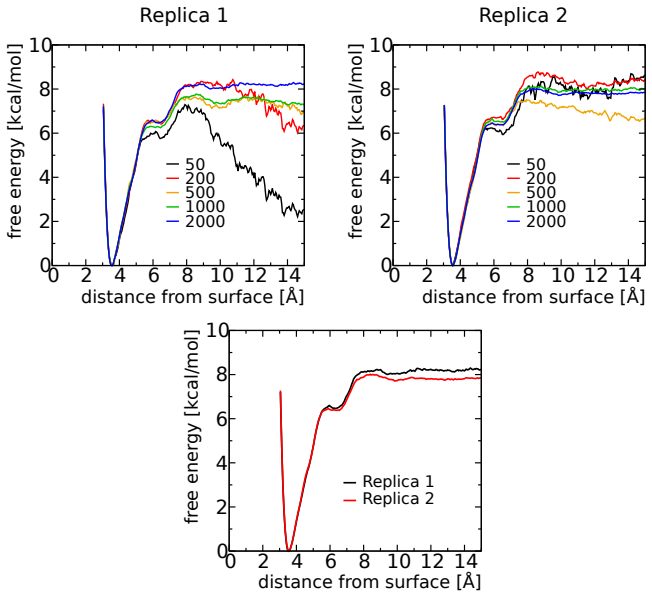


Fig. 10: Comparison of convergence in replicas.

(b) If the calculations are converging, we should see the two independent simulations predict PMFs that become more similar as the simulated time increases. Run the following script to compare the mean force estimates between the two simulations as a function of simulated time.

```
tclsh colvarsHistoryDev.tcl 0.04 ../fixed_replica{1,2}_hist/abf_3_15_graph_bPh.2.hist
xmgrace rep_dev.dat &
```

Here 0.04 is the time between frames of the `.hist.grad` file in ns (`historyFreq` × `timestep`. As shown below, we find that the after about 10 ns, the deviation between the two independent calculations falls like $\sim t^{-1/2}$, which is the expected convergence rate for a statistical process. Note that this implies that if you want to reduce the uncertainty in your calculations by a factor of 2, you need to run 4 times longer. Reducing it by a factor of 10 requires simulations 100 times longer.

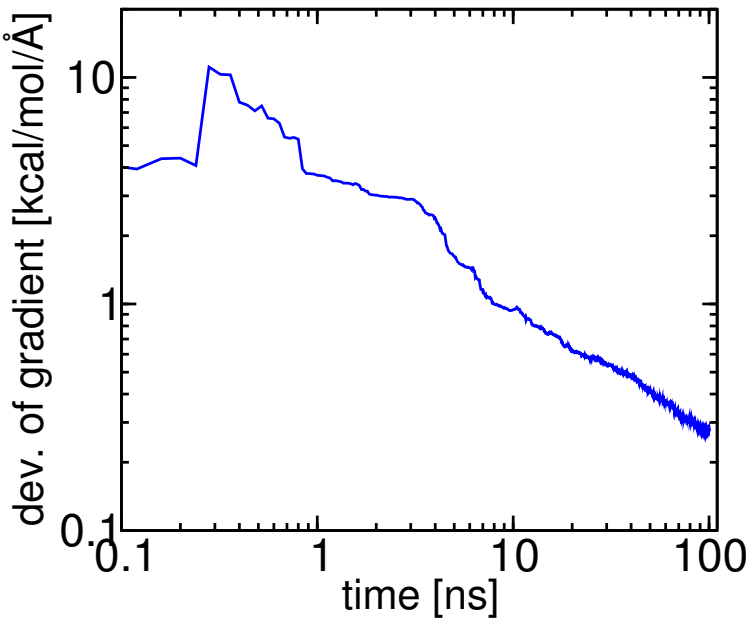


Fig. 11: Root-mean-square deviation of the estimated mean force between two independent ABF calculations as a function of simulated time.

(c) To further verify the results of adaptive biasing force, it is possible to compare with free energy perturbation calculations, we present the following plot as an example:

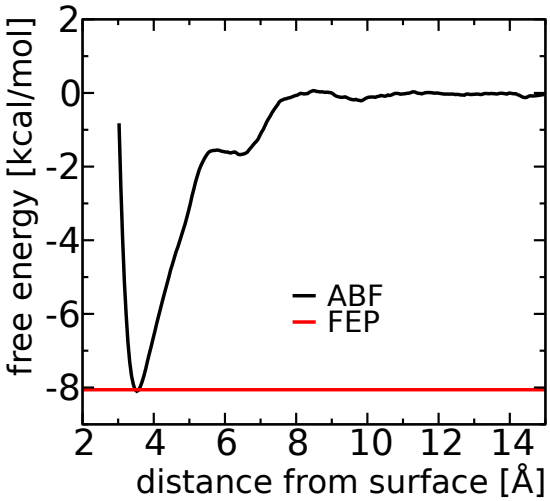


Fig. 12: Comparison of convergence of two free energy methods.

6 Comparing to Experiment

As is extensively explained by Comer et al. [1], it is possible to use the equation bellow to estimate the adsorption equilibrium constant (K_{eq}) from the simulations:

$$K_{\mathrm{eq}} = \frac{A_{\mathrm{NP}}}{M_{\mathrm{NP}}} \int_0^c \mathrm{d}z \, \exp\left[-\beta w(z)\right], \tag{2}$$

where $A_{\mathrm{NP}}/M_{\mathrm{NP}}$ is the specific surface area, which can be determined experimentally (for instance, by the Brunauer–Emmett–Teller method [19]) and $w(z)$ is the PMF determined from the simulations. When adsorption is strong, $w(z) \ll -k_{\mathrm{B}}T$, any reasonable choice of the cutoff c will yield essentially identical results (see the Supporting Information of Comer et al. [1].

This equation can be easily executed usin a AWK script as below:

```
awk -v dz=0.05 -v kT=0.59616124 -v ssa=233 '{s+=dz*exp(-$2/kT)};
 END {print 0.0001*ssa*s}' out_check_pmf.1.pmf > keq.out
```

Here, `dz` is the bin size and `ssa` is the specific surface area (experimentally determined). The 0.0001 is conversion factor to obtain the desired units, mL/g, from the specific surface area (in m^2/g) and the integral over `dz` (in Å). From this, we obtain the value 1.0×10^4 mL/g, which differs by more than an order of magnitude from the experimental value 4.8×10^5 [19, 20]. However, the situation is actually not as bad as it might seem. First, we have predicted this quantity to some accuracy using only our computer, with no need to perform costly or time-consuming experiments. Second, while the absolute equilibrium constant on graphene is underestimated for many small aromatic molecules, the simulations do an excellent job of predicting the ranking of equilbrium constants among different molecules [1]. Indeed, over a set of 29 small aromatic molecules, we obtain correlation coefficients $r \geq 0.90$ between the results derived from experiment and those derived from simulation. Such predictions are not easily obtained. For example, because graphene is a hydrophobic surface, one might naïvely consider that the molecules can be ranked by their hydrophobicity, which can be quantified by the octanol–water partition coefficient of each. This line of reasoning yields very poor results—a correlation coefficient of $r = 0.53$. For instance, both simulation and experiment predict that propylbenzene has a much lower affinity for graphene than 4-nitrotoluene, despite the fact that propylbenzene has a much greater octanol–water partition coefficient, and, thus, hydrophobicity.

The outcome of applying the technique described above to chemically distinct surfaces and other sets of small molecules will depend on the accuracy of force fields for these materials. Many existing force fields need to be better validated and perhaps improved to yield accurate predictions of adsorption affinities. However, the outlook for using molecular dynamics to rapidly predict affinity between small molecules and surfaces, enabling computational design, appears hopeful.

References

1. Comer J, Chen R, Poblete H, Vergara-Jaque A, Riviere JE (2015) Predicting adsorption affinities of small molecules on carbon nanotubes using molecular dynamics simulation. *ACS Nano* DOI: 10.1021/acsnano.5b03592.

2. Humphrey W, Dalke A, Schulten K (1996) VMD – Visual Molecular Dynamics. *J Mol Graphics* 14:33–38.

3. Aksimentiev A, Brunner R, Cohen J, Comer J, Cruz-Chu E, Hardy D, Rajan A, Shih A, Sigalov G, Yin Y, Schulten K (2008) Computer modeling in biotechnology: a partner in development. In *Nanostructure Design: Methods and Protocols*, volume 474 of *Methods in Molecular Biology*, pages 181–234. Humana Press. DOI: 10.1007/978-1-59745-480-3_11.

4. Comer J, Wells DB, Aksimentiev A (2011) Modeling nanopores for sequencing DNA. In *DNA Nanotechnology*, volume 749, chapter 22, pages 317–358. Humana Press.

5. Vanommeslaeghe K, Hatcher E, Acharya C, Kundu S, Zhong S, Shim J, Darian E, Guvench O, Lopes P, Vorobyov I, MacKerell AD (2010) CHARMM general force field: A force field for drug-like molecules compatible with the CHARMM all-atom additive biological force fields. *J Comput Chem* 31(4):671–690.

6. Vanommeslaeghe K, MacKerell Jr AD (2012) Automation of the CHARMM general force field (CGenFF) I: Bond perception and atom typing. *J Chem Inf Model* 52(12):3144–3154.

7. Vanommeslaeghe K, Raman EP, MacKerell Jr AD (2012) Automation of the CHARMM general force field (CGenFF) II: Assignment of bonded parameters and partial atomic charges. *J Chem Inf Model* 52(12):3155–3168.

8. Darve E, Pohorille A (2001) Calculating free energies using average force. *J Chem Phys* 115:9169–9183.

9. Comer J, Gumbart JC, Hénin J, Lelièvre T, Pohorille A, Chipot C (2015) The adaptive biasing force method: Everything you always wanted to know but were afraid to ask. *J Phys Chem B* 119(3):1129–1151.

10. Fiorin G, Klein ML, Hénin J (2013) Using collective variables to drive molecular dynamics simulations. *Mol Phys* 111(22-23):3345–3362.

11. Bussi G, Laio A, Parrinello M (2006) Equilibrium free energies from nonequilibrium metadynamics. *Phys Rev Lett* 96(9):90601.

12. Torrie G, Valleau J (1977) Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling. *J Comp Phys* 23(2):187–199.

13. Roux B (1995) The calculation of the potential of mean force using computer simulations. *Comp Phys Commun* 91:275–282.

14. Lelièvre T, Rousset M, Stoltz G (2008) Long-time convergence of an adaptive biasing force method. *Nonlinearity* 21(6):1155.

15. Lelièvre T, Minoukadeh K (2011) Long-time convergence of an Adaptive Biasing Force method: the bi-channel case. *Archive for rational mechanics and analysis* 202(1):1–34.

16. Phillips JC, Braun R, Wang W, Gumbart J, Tajkhorshid E, Villa E, Chipot C, Skeel RD, Kale L, Schulten K (2005) Scalable molecular dynamics with NAMD. *J Comput Chem* 26(16):1781–1802.

17. Feller SE, Zhang YH, Pastor RW, Brooks BR (1995) Constant pressure molecular dynamics simulations — the Langevin piston method. *J Chem Phys* 103:4613–4621.

18. Darden TA, York DM, Pedersen LG (1993) Particle mesh Ewald: An $n \log n$ method for Ewald sums in large systems. *J Chem Phys* 98:10089–10092.

19. Xia XR, Monteiro-Riviere NA, Mathur S, Song X, Xiao L, Oldenberg SJ, Fadeel B, Riviere JE (2011) Mapping the surface adsorption forces of nanomaterials in biological systems. *ACS Nano* 5(11):9074–9081.

20. Chen R, Zhang Y, Darabi Sahneh F, Scoglio CM, Wohlleben W, Haase A, Monteiro-Riviere NA, Riviere JE (2014) Nanoparticle surface characterization and clustering through concentration-dependent surface adsorption modeling. *ACS Nano* 8(9):9446–9456. DOI: 10.1021/nn503573s.