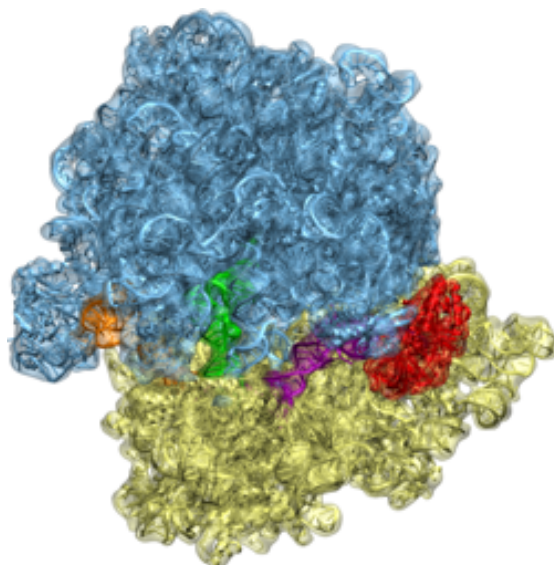


University of Illinois at Urbana-Champaign
NIH Resource for Macromolecular Modeling and Bioinformatics
Beckman Institute
Computational Biophysics Workshop

Molecular Dynamics Flexible Fitting



MDFF Developers:
Kwok-Yan Chan
Ryan McGreevy
Leonardo G. Trabuco
Elizabeth Villa

October 3, 2011

A current version of this tutorial is available at
<http://www.ks.uiuc.edu/Training/Tutorials/>

Contents

1	Overview of MDFF commands	5
2	A simple MDFF example	7
2.1	Generating a simulated density map	7
2.2	Converting the density map to an MDFF potential	9
2.3	Preparing the initial structure	10
2.4	Defining secondary structure restraints	11
2.5	Rigid-body docking the structure into the density map	12
2.6	Running the MDFF simulation with NAMD	13
2.7	Visualizing the MDFF trajectory	13
2.8	Calculating the root mean square deviation	14
2.9	Calculating the cross-correlation coefficient	16
3	MDFF with explicit solvent	18
3.1	Preparing the initial structure	18
3.2	Preparing the density map	20
3.3	Running the MDFF simulation	21
3.4	Analyzing the results	21
4	MDFF with Domain Restraints	23
4.1	Preparing the initial structure	23
4.2	Setting up the Domain PDB file	23
4.3	Running the MDFF simulation	24
4.4	Analyzing the results	26
5	MDFF with Symmetry Restraints	27
5.1	Preparing the initial structure	27
5.2	Setting up the Symmetry PDB file	27
5.3	Setting up the Transformation Matrix File	28
5.4	Running the MDFF simulation	28
5.5	Analyzing the results	29

Introduction

The molecular dynamics flexible fitting (MDFF) method can be used to flexibly fit atomic structures into density maps. The method was originally described in the manuscript:

Flexible fitting of atomic structures into electron microscopy maps using molecular dynamics. Leonardo G. Trabuco¹, Elizabeth Villa¹, Kakoli Mitra, Joachim Frank, and Klaus Schulten. *Structure*, 16:673-683, 2008.

We recommend reading the following practical guide before going through this tutorial:

Molecular dynamics flexible fitting: A practical guide to combine cryo-electron microscopy and x-ray crystallography. Leonardo G. Trabuco¹, Elizabeth Villa¹, Eduard Schreiner, Christopher B. Harrison, and Klaus Schulten. *Methods*, 49:174-180, 2009.

Required software

The necessary capabilities for setting up and analyzing MDFF simulations are implemented in VMD (Visual Molecular Dynamics), a molecular visualization and analysis program. MDFF simulations are performed using NAMD (NANoscale Molecular Dynamics), a molecular dynamics simulation program. Both VMD and NAMD are developed by the Theoretical and Computational Biophysics Group at the University of Illinois at Urbana-Champaign.

To apply the MDFF method you need to download and install both VMD and NAMD. Download and installation instructions can be found following the links above (if you are reading the electronic version of this document). In this tutorial we assume you are familiar with VMD; thus, we recommend that you complete the VMD tutorial beforehand. Completing the NAMD tutorial is not critical for understanding this tutorial, but it is nonetheless recommended. In this tutorial, we also use the third-party software package Situs for rigid-body docking (see Section 2.5 for details).

Tutorial Topics and Files

The tutorial starts with a brief overview of MDFF commands available in VMD (Section 1). In Section 2, a simple example of MDFF in vacuo is worked out. This first example uses two atomic structures of adenylate kinase in different conformations, and a simulated map is generated from one of the conformations, which is then used as a target for MDFF. All the basic steps for setting up, running, and analyzing MDFF simulations are covered. In Section 3, a similar MDFF simulation is performed, but this time in explicit solvent. In Section 4, use of domain restraints to maintain rigid domain during MDFF simulations is discussed. Finally, in Section 5, use of symmetry restraints for MDFF of

¹Equal contribution

symmetric molecules is covered. Other topics will be covered in a future version of this tutorial, including MDFF for RNA-containing systems, multi-step MDFF protocols, and interactive MDFF. The files provided with this tutorial are listed in the next table.

Table 1: Files provided for each section of this tutorial. All files can be found at the `mdff-tutorial-files` directory.

<hr/> <i>Section 2: A simple MDFF example</i> <hr/>
1ake-colores.pdb
1ake-initial.pdb
4ake-target.pdb
adk-step1-result.dcd
adk-step2-result.dcd
<hr/> <i>Section 3: MDFF with explicit solvent</i> <hr/>
adk-solvent-step1-result.dcd
adk-solvent-step2-result.dcd
<hr/> <i>Section 4: MDFF with Domain Restraints</i> <hr/>
acoasyn-initial.pdb
acoasyn-initial.psf
acoasyn-target.dx
acoasyn-target.pdb
domain-step1-result.dcd
no-domain-step1-result.dcd
par_all27_prot_lipid_na.inp
<hr/> <i>Section 5: MDFF with Symmetry Restraints</i> <hr/>
helix.pdb
helix.psf
helix-target.dx
helix-matrices.txt
no-symmetry-step1-result.dcd
par_all27_prot_lipid_na.inp
set_symmetry.tcl
symmetry-step1-result.dcd
<hr/>

1 Overview of MDFF commands

At present, all MDFF commands in VMD are available only through the Tcl command-line interface. You can use the text console called Tk Console to enter commands. You can also use the VMD prompt in the text console window. This window normally contains the prompt `vmd >`. For simplicity, in this tutorial we use several VMD text commands (not specific to MDFF) that are also available via the graphical interface.

- 1 Start a new VMD session. In the VMD Main menu select **Extensions** → **Tk Console** to open the VMD TkConsole window (Fig. 1). You can now start entering Tcl/Tk commands here.

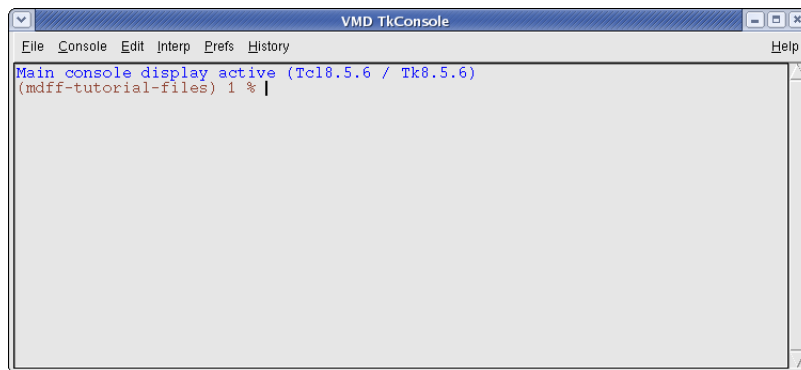


Figure 1: The VMD Tk Console window.

- 2 In order to use MDFF commands, you first need to load the MDFF package. This can be done by entering the following command in the Tk Console window:

```
package require mdff
```
- 3 Throughout the tutorial we assume that MDFF commands have been enabled by running the command above. You can add the line above to your VMD startup file `.vmdrc` or `vmd.rc` in Unix or Windows, respectively. The VMD startup file is typically located in the user's home directory. Check the VMD user guide for more information.

4 Type `mdff` to see a list of available MDFF commands. The following information should be printed in the console:

Usage: `mdff <command> [args...]`

Commands:

```
ccc          -- calculates the cross-correlation coefficient
check       -- monitors the fitting via RMSD and CCC
constrain   -- creates a pdb file for restraining atoms
delete      -- deletes volume corresponding to atomic structure
edges       -- creates a map with smooth edges
fix         -- creates a pdb file for fixing atoms
griddx      -- creates a map for docking
gridpdb     -- creates a pdb file with atomic masses in the
              beta field
setup       -- writes a NAMD configuration file for MDFF
sim         -- creates a simulated map from an atomic structure
```

5 You can type any command to obtain its usage information. For example, type `mdff sim` to check the syntax of the command that generates a simulated map from an atomic structure. We will cover most MDFF commands throughout the tutorial.

WARNING: *The MDFF software is considered experimental and the syntax of commands described in this tutorial is subject to change.*

2 A simple MDFF example

In a typical MDFF application, a high-resolution atomic structure of a macromolecule is flexibly fitted into a low-resolution density map of the same macromolecule imaged in a different conformation. As a first example, however, we will make use of two atomic structures representing different conformations of the protein adenylate kinase. The PDB file `1ake-initial.pdb` will provide the initial structure, and a target density map will be generated from the PDB file `4ake-target.pdb`. You will then flexibly fit the initial structure into the target density map using MDFF. You can compare the two structures by loading the PDB files in VMD (Fig. 2).

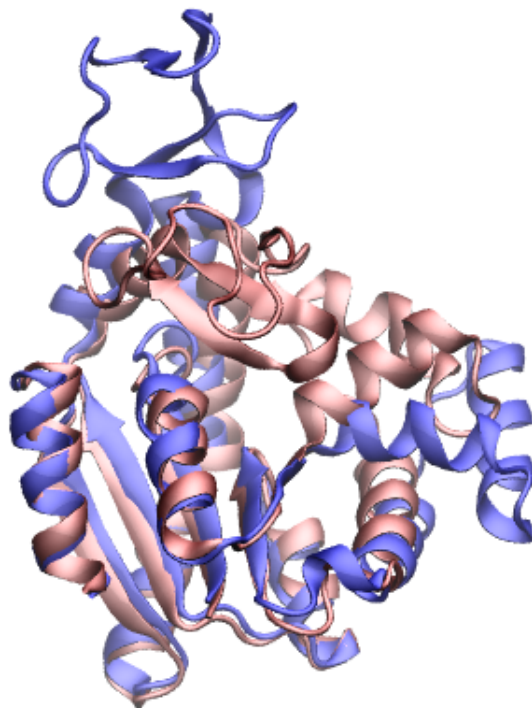


Figure 2: Initial and target structures shown in red and blue, respectively.

2.1 Generating a simulated density map

You will first generate a noise-free, simulated electron microscopy map from the PDB file `4ake-target.pdb`. The simulated map is generated by interpolating the atomic number of each atom onto a 3-D grid and low-pass filtering it to the desired resolution.

- 1 In the VMD Tk Console, load the target PDB file by typing:

```
mol new 4ake-target.pdb
```
- 2 To make sure the target structure is complete, we will use the VMD plugin AutoPSF, which automates building structures for molecular dynamics simulations. In the VMD Main window, choose Extensions → Modeling → Automatic PSF Builder (Fig. 3).

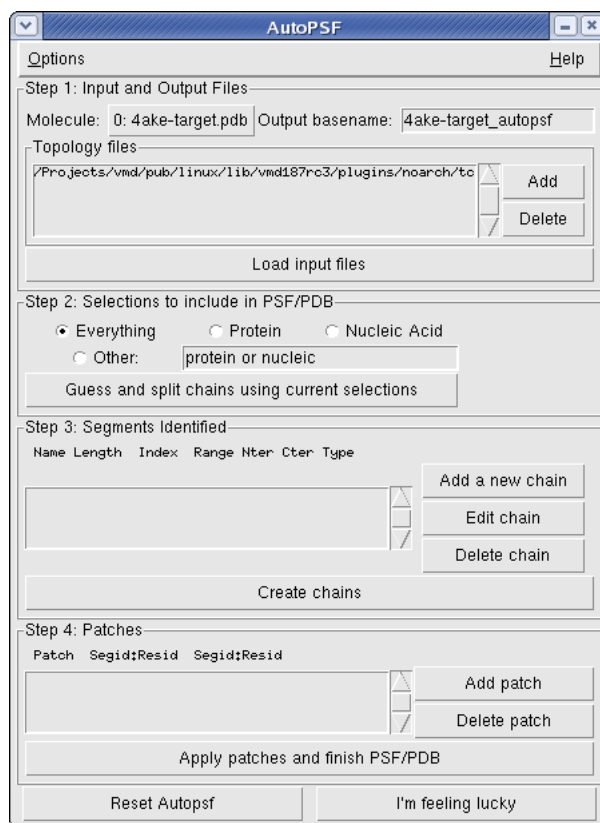


Figure 3: The VMD AutoPSF window.

- 3 Click on **Load input files** to load the topology information from the CHARMM force field. Then click on **Guess and split chains using current selections**, and finally click on **Create chains**. Hit OK in the two information boxes that will appear. The files generated by AutoPSF (`4ake-target_autopsf.psf` and `4ake-target_autopsf.pdb`) will be automatically loaded in VMD as the top molecule. Note that in some other cases, patches may need to be added as identified in the Step 4 box. In such cases, click **Apply patches**

and finish PSF/PDB to add the patches and new PSF/PDB files will be generated and loaded.

- 4 Generate a simulated map using the `mdff sim` command. First, type `mdff sim` to get the usage information.

- 5 The usage information shows that `mdff sim` requires an atom selection, which can be created using the `atomselect` command:

```
set sel [atomselect top all]
```

- 6 The variable `sel` now contains a selection of all the atoms from the top molecule, which is usually the last molecule we loaded in VMD. To generate a simulated map with a resolution of 5 Å in, e.g., the Situs file format, type:

```
mdff sim $sel -res 5 -o 4ake-target_autopsf.situs
```

- 7 Load the simulated map in VMD by typing

```
mol new 4ake-target_autopsf.situs
```

You can change the appearance of the density map using the Graphics → Graphical Representations window. For example, choose Wireframe instead of Points for rendering the volume. You can choose the contour level by sliding the Isovalue bar. Fig. 4 shows both the target structure and the simulated map.

2.2 Converting the density map to an MDFF potential

Now that you have a density map, you need to convert it to the potential U_{EM} used by MDFF, where

$$U_{EM}(\mathbf{R}) = \sum_j w_j V_{EM}(\mathbf{r}_j), \quad (1)$$

and

$$V_{EM}(\mathbf{r}) = \begin{cases} \xi \left[1 - \frac{\Phi(\mathbf{r}) - \Phi_{thr}}{\Phi_{max} - \Phi_{thr}} \right] & \text{if } \Phi(\mathbf{r}) \geq \Phi_{thr}, \\ \xi & \text{if } \Phi(\mathbf{r}) < \Phi_{thr}. \end{cases} \quad (2)$$

Here, Φ denotes the values of the density map at each grid point. The threshold value Φ_{thr} is used for flattening the solvent density by clamping density values below the threshold. By default, MDFF uses $\Phi_{thr} = 0$, which is usually where the solvent peak lies in cryo-EM maps. In this example using a simulated map, however, there is no solvent contribution to the density. The scaling factors w_j and ξ will be discussed in Sections 2.3 and 2.6, respectively.

MDFF takes advantage of the NAMD’s `gridForces` feature, through which an arbitrary external potential defined on a 3-D grid can be added to a molecular dynamics simulation (D. Wells, V. Abramkina, and A. Aksimentiev, *J. Chem. Phys.*, **127**:125101-125110, 2007). NAMD’s `gridForces` feature requires an input 3-D map defining the external potential in the DX file format.

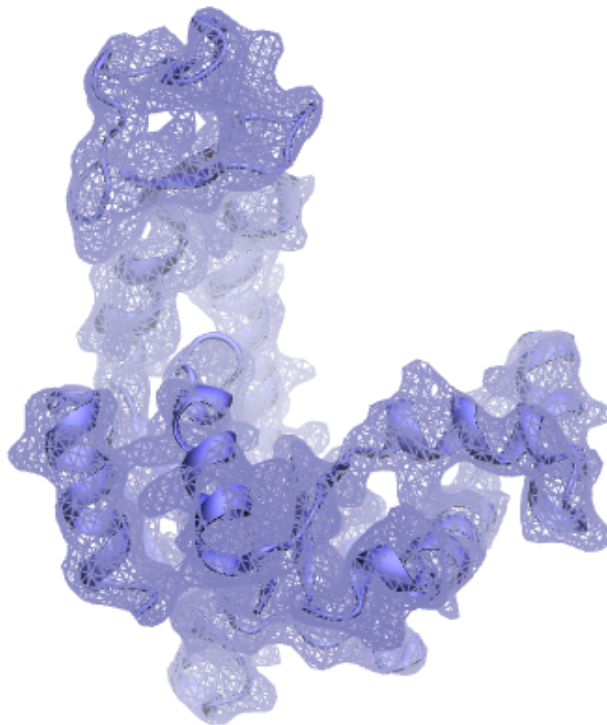


Figure 4: Target structure and simulated map shown in cartoon and mesh, respectively.

- 1 To obtain a 3-D map defining V_{EM}/ξ , which will be provided to NAMD, you need to use the command `mdff griddx`. The scaling factor ξ will be provided to NAMD separately, so the same DX map can be used with different scaling factors ξ . To obtain the usage information, simply type `mdff griddx`.
- 2 Run the following command to generate the DX file defining V_{EM}/ξ :

```
mdff griddx -i 4ake-target_autopsf.situs -o  
4ake-target_autopsf-grid.dx
```

2.3 Preparing the initial structure

Now we turn to the preparation of the input structure. The steps required for setting up a regular molecular dynamics simulation are also required by MDFF.

- 1 Load the initial structure in VMD by typing:

```
mol new 1ake-initial.pdb
```

- 2 In order to complete the structure and generate a PSF file containing all the connectivity information and partial atomic charges required by NAMD, use again the AutoPSF plugin as in Section 2.1. If you are working on the same VMD session from the beginning of the tutorial, make sure you click the Reset AutoPSF button and then choose the correct molecule in the AutoPSF plugin. You should be able to generate the files `lake-initial.autopsf.psf` and `lake-initial.autopsf.pdb`.
- 3 You now need to generate a PDB file containing the per-atom scaling factors w_j in Equation 1, which are set to the atomic mass by the `mdff gridpdb` command. As usual, check the usage by typing `mdff gridpdb` with no arguments.
- 4 Now run the command


```
mdff gridpdb -psf lake-initial.autopsf.psf
-pdb lake-initial.autopsf.pdb -o lake-initial.autopsf-grid.pdb
```

2.4 Defining secondary structure restraints

We will apply restraints during the MDFF simulation to enforce the secondary structure of our protein. NAMD's `extraBonds` feature allows for additional bonds, angles, dihedral angles, and impropers to be defined. The VMD plugin `ssrestraints` automates the generation of `extraBonds` input files that define secondary structure restraints.

- 1 First, load the `ssrestraints` plugin by typing


```
package require ssrestraints
```
- 2 Type `ssrestraints` to check the usage information.
- 3 Define restraints for ϕ and ψ dihedral angles for amino acid residues in helices or sheets, as well as restraints for hydrogen bonds involving backbone atoms from the same residues:


```
ssrestraints -psf lake-initial.autopsf.psf
-pdb lake-initial.autopsf.pdb -o lake-extrabonds.txt -hbonds
```

MDFF may result in relatively large forces being applied to certain atoms, which could in turn lead to certain structural artifacts such as chiral centers with wrong handedness or generation of *cis* peptide bonds. This is a limitation of any modeling technique based on commonly used molecular dynamics force fields that do not define explicit terms to prevent such structures. VMD provides two plugins to detect, fix, and prevent generation of *cis* peptide bonds and chirality errors. Please refer to the Structure Check tutorial for more details.

- 4 Here we will simply restrain peptide bonds to their current *cis/trans* configuration, as well as all chiral centers to their current handedness. First,

make sure the initial structure generated by AutoPSF is loaded as the top molecule in VMD. If not, you can load it by running:

```
mol new lake-initial_autopsf.psf
mol addfile lake-initial_autopsf.pdb
```

- 5 Use the cispeptide plugin to restrain cis peptide bonds to their current cis/trans configuration:

```
cispeptide restrain -o lake-extrabonds-cispeptide.txt
```

- 6 Analogously, use the chirality plugin to restrain chiral centers to their current handedness:

```
chirality restrain -o lake-extrabonds-chirality.txt
```

2.5 Rigid-body docking the structure into the density map

Prior to performing flexible fitting of the atomic structure into the density map, it is necessary to perform a rigid-body docking. There are a few software packages that provide this functionality. Here we will use the Situs package, which we assume you have installed in your system. In case you want to proceed working through the MDFF tutorial before installing Situs, we also provide the rigid-body docking results (jump to step 5 below).

- 1 Download and install the Situs package from <http://situs.biomachina.org/>.

- 2 We will use the `colores` program from the Situs package, which implements an exhaustive, FFT-accelerated 6-D search. For more information, please refer to the correlation-based docking tutorial on the Situs website.

- 3 Assuming the `colores` program is in your path, run on a terminal:

```
colores 4ake-target_autopsf.situs lake-initial_autopsf.pdb
-res 5 -nopowell
```

In the interest of time, we disabled Powell optimization with the `nopowell` option above. In a real application, we recommend you don't use this option. You can also improve the quality of the rigid-body docking by, e.g., decreasing the sampling step (`-deg` option). This step should take about 10 minutes on a single processor.

- 4 Situs will generate a number of PDB files corresponding to rotations and translations of the initial structure. These files are called `col_best_###.pdb`, where `###` is a number. You should typically choose `col_best_001.pdb` as the starting structure for MDFF, but first you should visually inspect the files generated by `colores` by loading them in VMD, together with the density map. Rename the file `col_best_001.pdb` to `lake-initial_autopsf-docked.pdb`.

- 5 In case you don't have access to Situs while working through this tutorial, or if you don't want to wait for `colores` to complete, we provide the resulting rigid-body docked structure in the file `lake-colores.pdb`. Copy this file to `lake-initial_autopsf-docked.pdb`.

2.6 Running the MDFF simulation with NAMD

As mentioned above, MDFF simulations are performed with the program NAMD, which we assume you have installed in your system.

- 1 You must first generate a NAMD configuration file. This is automated by the MDFF plugin in VMD. In the VMD Tk Console window, type `mdff setup` for usage information.
- 2 Generate a NAMD configuration file using the command:

```
mdff setup -o adk -psf lake-initial_autopsf.psf
          -pdb lake-initial_autopsf-docked.pdb
          -griddx 4lake-target_autopsf-grid.dx
          -gridpdb lake-initial_autopsf-grid.pdb
          -extrab {lake-extrabonds.txt lake-extrabonds-cispeptide.txt
                  lake-extrabonds-chirality.txt} -gscale 0.3 -numsteps 50000
```

Most options above specify the names of files you generated in previous steps and should be self-explanatory. The option `-gscale` defines the scaling factor ξ in Equation 1. In the interest of time, we chose to run for only 50 ps (`-numsteps` option). Real MDFF applications will typically require longer simulations to reach convergence. You will learn how to judge convergence of an MDFF simulation in Sections 2.8 and 2.9.

- 3 Generate a second NAMD configuration file in which only energy minimization will be performed with a much higher scaling factor ξ :

```
mdff setup -o adk -psf lake-initial_autopsf.psf
          -pdb lake-initial_autopsf-docked.pdb
          -griddx 4lake-target_autopsf-grid.dx
          -gridpdb lake-initial_autopsf-grid.pdb
          -extrab {lake-extrabonds.txt lake-extrabonds-cispeptide.txt
                  lake-extrabonds-chirality.txt} -gscale 10 -minsteps 2000
          -numsteps 0 -step 2
```

- 4 Quit VMD.
- 5 Run NAMD using the configuration file generated by VMD, i.e., run the following commands in a terminal:

```
namd2 adk-step1.namd > adk-step1.log
namd2 adk-step2.namd > adk-step2.log
```

This step should take about 35 minutes on a single processor. If you don't want to wait, you can proceed to the next step and use the provided trajectory files, as explained in the next section.

2.7 Visualizing the MDFF trajectory

The resulting trajectories will be saved to files `adk-step1.dcd` and `adk-step2.dcd`. If you want to continue working through the tutorial before the simulations are

completed, you can use the provided trajectory files `adk-step1-result.dcd` and `adk-step2-result.dcd` instead. Please note that, due to the stochastic nature of molecular dynamics simulations, it is expected that the trajectories obtained will differ from the ones provided. You will now load the trajectory files in VMD and visualize the MDFF results.

- 1 Start a new VMD session and load the target structure (for reference) by typing the following commands on the Tk Console:

```
mol new 4ake-target_autopsf.psf
mol addfile 4ake-target_autopsf.pdb
```

- 2 Open the Graphical Representations window available under Graphics → Representations and change the Drawing Method to `NewCartoon`. Change also the Coloring Method to `ColorID 0 blue`.

- 3 Open the Color Controls window available under Graphics → Colors. Change the background color to white by clicking on `Display, Background`, and finally `8 white`.

- 4 Load the initial structure by typing the following commands on the Tk Console:

```
mol new 1ake-initial_autopsf.psf
mol addfile 1ake-initial_autopsf-docked.pdb
```

- 5 Following the same steps as above, change the representation to `NewCartoon` and the color to red. Your VMD OpenGL Display should look similar to Fig. 2.

- 6 Load the MDFF trajectories using the following commands:

```
mol addfile adk-step1.dcd
mol addfile adk-step2.dcd
```

- 7 You can navigate through the trajectory using the VMD Main window. For example, you can drag the trajectory slider to jump to any frame in the trajectory.

2.8 Calculating the root mean square deviation

The easiest way to track the evolution of the MDFF simulation and track its convergence is by plotting the root mean square deviation (RMSD) over time. There are different interfaces for RMSD calculation in VMD. For example, the `mdff check` command can be used to take a quick look at the RMSD evolution of the simulated system.

- 1 Type `mdff check` on the Tk Console window for usage information.

- 2 Plot the backbone RMSD with respect to the initial structure for each trajectory frame using the command

```
mdff check -rmsd
```

A window similar to the one depicted in Fig. 5 should appear. Note how the RMSD levels off toward the end of the simulation.

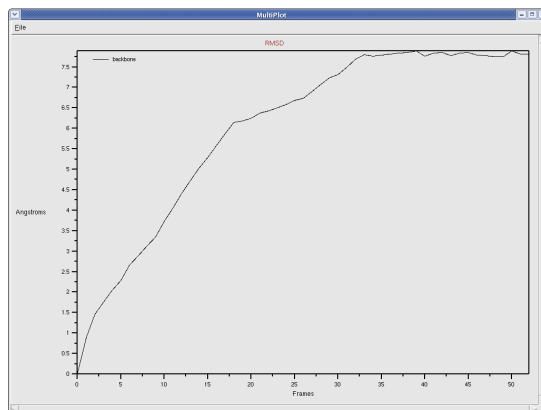


Figure 5: Backbone RMSD with respect to the initial structure for the MDFF simulation.

- 3 Now plot the backbone RMSD with respect to the target structure using the command

```
mdff check -rmsd -refpdb 4ake-target_autopsf.pdb
```

A window similar to the one depicted in Fig. 6 should now appear. Note how the RMSD decreases (the fitting improves) in the last couple of frames, which correspond to the energy minimization performed in the MDFF step 2.

- 4 Now let's see how you can use Tcl commands to calculate RMSDs. You will first create atom selections for both the initial and the target structures. If you followed the previous steps exactly the target structure should be loaded as molecule 0, whereas the initial structure and MDFF trajectory should be loaded as molecule 1. Type the following commands:

```
set selbbref [atomselect 0 "backbone"]
set selbb [atomselect 1 "backbone"]
```

- 5 Make sure the atom selection `selbb` points to the initial frame loaded, which is the initial structure:

```
$selbb frame 0
```

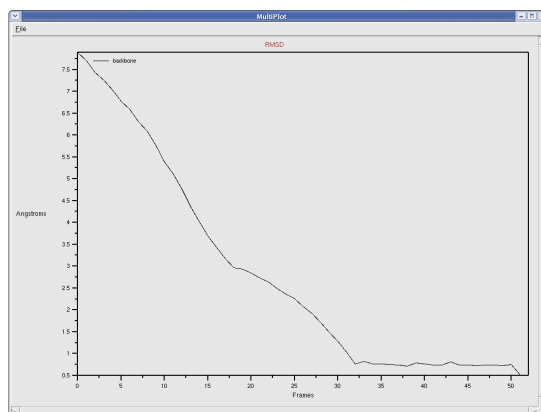


Figure 6: Backbone RMSD with respect to the target structure for the MDFF simulation.

- 6 Now calculate the RMSD between `selbb` and `selbbref`:

```
measure rmsd $selbb $selbbref
```

You should get an RMSD of 7.89 Å.

- 7 Now make the atom selection `selbb` point to the last frame of the trajectory and recalculate the RMSD:

```
$selbb frame last
```

```
measure rmsd $selbb $selbbref
```

If you used the provided trajectory files, you should get an RMSD of 0.49 Å. The RMSD value will naturally vary if you load a different MDFF trajectory.

2.9 Calculating the cross-correlation coefficient

Now you will calculate the cross-correlation coefficient (CCC) between the target density map and each frame of the MDFF trajectory. Internally, a simulated map is created from the atomic structure and the cross-correlation coefficient between the two maps is calculated.

- 1 Plot the CCC between each MDFF trajectory frame and the target map using the command

```
mdff check -ccc -map 4ake-target_autopsf.situs -res 5
```

This step should take about 1.5 minutes and will be much faster in a future version of MDFF. A window similar to the one depicted in Fig. 7 should appear. Note how the CCC follows the same trend as the RMSD previously inspected.

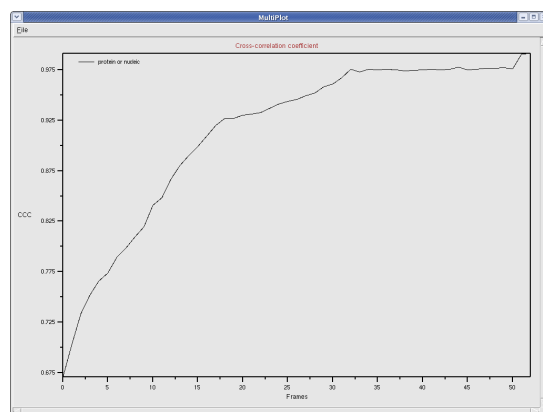


Figure 7: Cross-correlation coefficient between target density map and each frame of the MDFF simulation.

- 2 The command `mdff ccc` can be used to calculate the CCC between an atom selection and a density map. First, create atom selections containing all the atoms, similarly to when you calculated RMSDs:

```
set selallref [atomselect 0 "all"]
set selall [atomselect 1 "all"]
```

- 3 Type `mdff ccc` to obtain the usage information.

- 4 Calculate the initial CCC:

```
$selall frame 0
mdff ccc $selall -i 4ake-target_autopsf.situs -res 5
```

You should get a CCC of 0.671.

- 5 Now calculate the final CCC:

```
$selall frame last
mdff ccc $selall -i 4ake-target_autopsf.situs -res 5
```

If you used the provided trajectory files, you should get a CCC of 0.991.

- 6 Quit VMD.

3 MDFF with explicit solvent

In the last section you learned how to set up a simple MDFF simulation in vacuo. Now you will learn how to set up a similar simulation in explicit solvent.

3.1 Preparing the initial structure

We will start with the structure that was already prepared in the previous section, i.e., files `lake-initial_autopsf.psf` and `lake-initial_autopsf-docked.pdb`.

- 1 Start a new VMD session.
- 2 Load the initial structure you prepared for MDFF in vacuo in the previous section:

```
mol new lake-initial_autopsf.psf
mol addfile lake-initial_autopsf-docked.pdb
```

- 3 Embed this structure into a water box using the `solvate` plugin. In the VMD Main Window, choose `Extensions` → `Modeling` → `Add Solvation Box` (Fig. 8). Set the box padding to 20 Å for maximum y, 5 Å for minimum y, and 10 Å for the remainder dimensions, as shown in the figure and click on `Solvate`. It is important that the target map falls completely within the water box, which is why we chose a larger padding in one of the dimensions. VMD will generate the files `solvate.psf` and `solvate.pdb`, which will be automatically loaded upon completion of this step. Load the target map (from the previous section) to visually ensure that the density falls completely within the water box (Fig. 9).

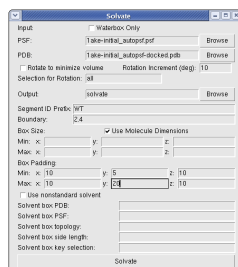


Figure 8: The VMD Solvate window.

- 4 For MD simulations in explicit solvent, it is usually desirable to have a neutrally charged system. This can be achieved by adding neutralizing counterions to the simulation system. One can also add additional ions to mimic in vivo or in vitro conditions. In this example, we will simply neutralize the system by adding either Na^+ or Cl^- ions using the `autoionize`

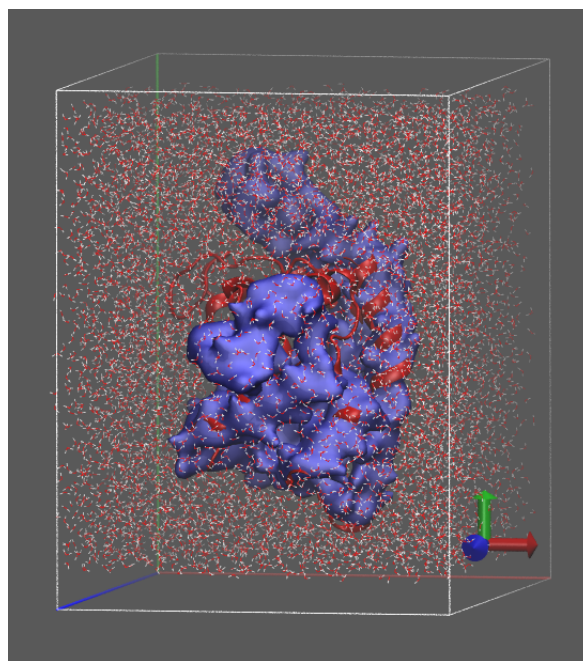


Figure 9: By loading the target density map you can visually ensure that the target density falls within the boundaries of the water box. If part of the target density is outside the water box, you need to adjust the padding accordingly and regenerate the water box.

plugin. In the VMD Main Window, choose Extensions → Modeling → Add Ions (Fig. 10). Uncheck the button defining the ion concentration, leaving only neutralization active, as shown in the figure. Click on Autoionize. VMD will generate the files `ionized.psf` and `ionized.pdb`.

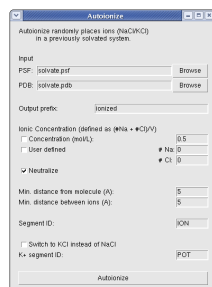


Figure 10: The VMD Autoionize window.

- 5 Generate a PDB file containing the per-atom scaling factors w_j in Equation 1, as in the previous section. By default, water molecules and ions are not coupled to the target map, i.e., they can equilibrate freely according to the MD force field and don't experience additional forces from MDFF:

```
mdff gridpdb -psf ionized.psf -pdb ionized.pdb
-o ionized-grid.pdb
```

- 6 Generate secondary structure restraints as in the previous section:

```
package require ssrestraints
ssrestraints -psf ionized.psf -pdb ionized.pdb
-o ionized-extrabonds.txt -hbonds
```

- 7 Generate restraints to prevent cis/trans peptide transitions and chirality errors:

```
mol new ionized.psf
mol addfile ionized.pdb
cispeptide restrain -o ionized-extrabonds-cispeptide.txt
chirality restrain -o ionized-extrabonds-chirality.txt
```

3.2 Preparing the density map

As you can see in Fig. 9, we already ensured that the target density map (blue) is completely within the water box, which is a requirement for MDFF in explicit solvent. However, the current implementation of NAMD's `gridforces` feature also requires that the entire target map be within the water box, which is clearly not the case (Fig. 9). To address this limitation, we will trim the target map so that it lies completely within the water box.

- 1 Trim the resampled map in all dimensions by a few of Angstroms to ensure it will be within the water box during the simulation. The `volutil` plugin provides some features for manipulating volumetric maps. To trim the target map by 7 Å in all dimensions, run:

```
package require volutil
volutil 4ake-target_autopsf-grid.dx -trim 7
-o 4ake-target_autopsf-grid-trimmed.dx
```

NOTE: *If the target map and the box are rotated with respect to each other, it may be necessary to resample the target map into a cell that fits completely inside the water box, which is not yet supported by VMD.*

- 2 Visualize in VMD the water box and the newly trimmed map to verify that it falls completely within the water box. Also ensure that the target macromolecular volume is contained in the new map. Load the trimmed map with the command:

```
mol new 4ake-target_autopsf-grid-trimmed.dx
```

3.3 Running the MDFF simulation

Generate NAMD configuration files similarly to the previous section.

- 1 For MDFF simulations in solvent we need to define periodic boundary conditions. We also use a different method to calculate electrostatic interactions that is more appropriate for this kind of simulation. All of this is taken care of by providing the extra option `-pbc` to `mdff setup`:

```
mdff setup -pbc -o adk-solvent -psf ionized.psf
-pdb ionized.pdb
-griddx 4ake-target_autopsf-grid-trimmed.dx
-gridpdb ionized-grid.pdb
-extrab {ionized-extrabonds.txt ionized-extrabonds-cispeptide.txt
ionized-extrabonds-chirality.txt} -gscale 0.3 -numsteps 100000
```

Note that we requested a simulation twice as long as in the previous section, since explicit-solvent MDFF simulations typically take longer to converge.

- 2 Once again, generate a second NAMD configuration file in which only energy minimization will be performed with a much higher scaling factor ξ :

```
mdff setup -pbc -o adk-solvent -psf ionized.psf
-pdb ionized.pdb
-griddx 4ake-target_autopsf-grid-trimmed.dx
-gridpdb ionized-grid.pdb
-extrab ionized-extrabonds.txt ionized-extrabonds-cispeptide.txt
ionized-extrabonds-chirality.txt} -gscale 10
-minsteps 2000 -numsteps 0 -step 2
```

- 3 Quit VMD.

- 4 Run NAMD using the configuration files generated by VMD, i.e., run the following commands in a terminal (or submit them to a cluster):

```
namd2 adk-solvent-step1.namd > adk-solvent-step1.log
namd2 adk-solvent-step2.namd > adk-solvent-step2.log
```

This step should take about 40 minutes on a cluster with 48 processors. If you don't want to wait, you can proceed to the next step and use the provided trajectory files, as explained in the next section.

3.4 Analyzing the results

The resulting trajectories will be saved to files `adk-solvent-step1.dcd` and `adk-solvent-step2.dcd`. If you want to continue working through the tutorial before the simulations are complete, you can use the provided trajectory files `adk-step1-result.dcd` and `adk-step2-result.dcd` instead. Once again, please note that due to the stochastic nature of molecular dynamics simulations

it is expected that the trajectories obtained will differ from the ones provided. As in the previous section, load the resulting trajectory, as well as the target structure, and repeat the analysis of the RMSD and CCC. Did the use of explicit solvent improve the MDFP results in this particular case?

4 MDFF with Domain Restraints

This section will show you how to set up MDFF simulations with domain restraints in `vaccum`. Domain restraints apply harmonic forces to user defined groups of atoms to maintain a rigid domain during MDFF simulations.

4.1 Preparing the initial structure

For this example we will be using a protein called Acetyl CoA Synthase, found in `acoasyn-initial.psf` and `acoasyn-initial.pdb`. The structure has already been rigid-body docked to a simulated density map of the protein in a different conformation. Refer to the section 2 for information on generating a simulated density map and rigid-body docking.

1 Start a new VMD session.

2 Generate a PDB file containing the per-atom scaling factors w_j in Equation 1, as in the previous section.

```
mdff gridpdb -psf acoasyn-initial.psf -pdb acoasyn-initial.pdb
-o acoasyn-grid.pdb
```

3 Generate secondary structure restraints as in the previous section:

```
package require ssrestraints
ssrestraints -psf acoasyn-initial.psf -pdb acoasyn-initial.pdb
-o acoasyn-extrabonds.txt -hbonds
```

4 Generate restraints to prevent cis/trans peptide transitions and chirality errors:

```
mol new acoasyn-initial.psf
mol addfile acoasyn-initial.pdb
cispeptide restrain -o acoasyn-extrabonds-cispeptide.txt
chirality restrain -o acoasyn-extrabonds-chirality.txt
```

4.2 Setting up the Domain PDB file

In order to define the domains that will be restrained, we must make a PDB file containing our designations. Domain restraints based on the Targeted MD (TMD) function of NAMD, so we will be setting up our restraints as a TMD run. For more information on TMD please read the NAMD documentation.

1 Load the pdb file we are using for the initial structure

```
mol new acoasyn-initial.psf
mol addfile acoasyn-initial.pdb
```

- 2 Select each group of atoms and set their beta column to the proper domain designation

```
set sel [atomselect top "all"]
$sel set beta 0
$sel set occupancy 0
set sel1 [atomselect top "segname 1 and name CA"]
$sel1 set beta 1
$sel1 set occupancy 1
set sel2 [atomselect top "segname 2 and name CA"]
$sel2 set beta 2
$sel2 set occupancy 1
$sel writepdb domain.pdb
```

Placing a 1 in the occupancy column lets NAMD know that these atoms should be involved in TMD. Atoms with the same beta value (>0) are treated as a single domain. Here we have assigned two domains (beta 1 and 2) which will be kept rigid during the MDFF simulations. These domains are shown using VMD in Fig. 11.

4.3 Running the MDFF simulation

Generate NAMD configuration files similarly to the first example. We will be using the density map provided, acoasyn-target.dx. For information regarding generating simulated density maps, please see the first MDFF tutorial section 2.

- 1 Generate a NAMD configuration file:

```
mdff setup -o domain -psf acoasyn-initial.psf
-pdb acoasyn-initial.pdb
-griddx acoasyn-target.dx
-gridpdb acoasyn-grid.pdb
-extrab {acoasyn-extrabonds.txt acoasyn-extrabonds-cispeptide.txt
acoasyn-extrabonds-chirality.txt} -gscale 1.0 -minsteps 2000
-numsteps 70000
```

- 2 Now we need to edit the configuration file domain-step1.namd that we just created by adding in the TMD parameters necessary for domain restraints. Open the file in any text editor and add the following lines anywhere:

```
tmd on
tmdfile domain.pdb
tmdk 500.
tmdfirststep 2001
tmdlaststep 72000
tmdoutputfreq 1000
```

These parameters turn TMD on and let NAMD know where to look for

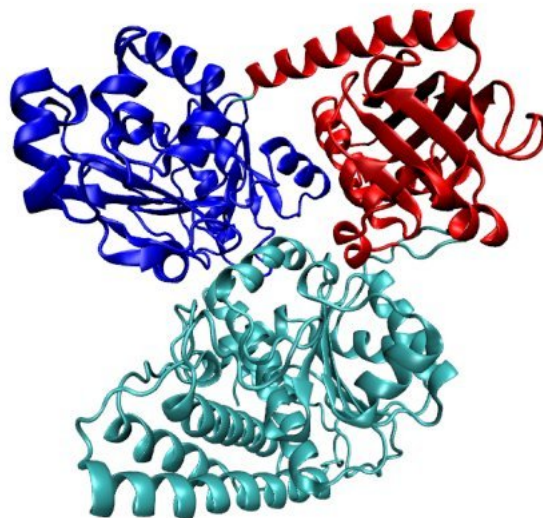


Figure 11: The two domains defined by the pdb file. Domain 1 in red and domain 2 in blue.

the domain information. `tmdk` is a constant which scales the harmonic force applied by the restraint on a domain and this constant is scaled down by the number of atoms in a domain. One can specify a per-atom force constant that assign force constant to individual atoms which is not discussed here. More information about these parameters can be found in the TMD documentation in the NAMD User's Guide.

3 Quit VMD.

4 Run NAMD using the configuration file generated by VMD, i.e., run the following command in a terminal:

```
namd2 domain-step1.namd > domain-step1.log
```

This step should take about 1.5 hours on a single desktop processor core. If you don't want to wait, you can proceed to the next step and use the provided trajectory files, as explained in the next section.

4.4 Analyzing the results

The resulting trajectory will be saved to the file `domain-step1.dcd`. If you want to continue working through the tutorial before the simulation is complete, you can use the provided trajectory file `domain-step1-result.dcd` instead. As in the previous sections, load the trajectory files and target structure `acoasyn-target.pdb` into VMD and repeat the RMSD and CCC analysis. Now try running the same simulation without any restraints by turning Targeted MD off by switching `tmd on` in the configuration file to `tmd off` and changing the output name to `no-domain`. Again, you can continue the tutorial by using the provided trajectory `no-domain-step1-result.dcd`. Load the trajectory and target structure into VMD and repeat the RMSD and CCC analysis. Also, look at the trajectories and comparing the cases with and without restraints.

5 MDFF with Symmetry Restraints

This section will show you how to set up a MDFF run using symmetry restraints. Symmetry restraints use harmonic forces to maintain a symmetric structure during MDFF simulations for symmetric molecules. The symmetric structure is determined by transforming and overlapping the atomic coordinates of all symmetric units and calculating the average positions of the transformed atoms.

5.1 Preparing the initial structure

For this example we will be using a nitrilase structure with helical symmetry, found in `helix.psf` and `helix.pdb`. The structure is already rigid-body docked into an experimental map (EMDB 1313) with Situs. The map has been converted into the 3D potential for MDFF, named as `helix-target.dx`. Please refer to previous section for use of Situs for rigid-body docking and use of `mdff griddx` for map conversion.

1 Start a new VMD session.

2 Generate a PDB file containing the per-atom scaling factors w_j in Equation 1, as in previous section.

```
mdff gridpdb -psf helix.psf -pdb helix.pdb
-o helix-grid.pdb
```

3 Generate secondary structure restraints as in previous section:

```
package require ssrestraints
ssrestraints -psf helix.psf -pdb helix.pdb
-o helix-extrabonds.txt -hbonds
```

4 Generate restraints to prevent cis/trans peptide transitions and chirality errors:

```
mol new helix.psf
mol addfile helix.pdb
cispeptide restrain -o helix-extrabonds-cispeptide.txt
chirality restrain -o helix-extrabonds-chirality.txt
```

5.2 Setting up the Symmetry PDB file

We need to create a symmetry PDB file containing the designations of symmetric units, i.e. what the symmetric units are. In the symmetry PDB file, the occupancy column denotes the "symmetry group" atoms belong to, while the beta column denotes the symmetric units designation. In this example, we have a single symmetry relationship, the helical symmetry, so we only have one symmetry group and hence the occupancy column is set to 1. We have nine different symmetric unit within this helical symmetry group, so the beta column of the first symmetric unit is set to 1 and increases by 1 for the next symmetric until the last symmetric unit with beta column assigned to 9. For more information on symmetry restraint parameters, please read the documentation.

- 1 Load the pdb file we are using for the initial structure

```
mol new helix.pdb
```

- 2 Assign beta and occupancy values for different symmetric units according to the rules described above. A tcl script has been provided to you for setting these values. Note that we are applying symmetric restraints to C_α atoms only.

```
source set_symmetry.tcl
```

5.3 Setting up the Transformation Matrix File

Next we have to create a matrix file which contains the transformation matrices needed to overlap the symmetric subunits based on their symmetry. If the matrix file is not given to NAMD, NAMD will attempt to generate these matrices automatically by guessing the symmetry information among the symmetric units. This file follows a specific format outlined below:

- 1 Matrices should be in order of symmetric units designation (beta column of the symmetry pdb file generated above) e.g. The first matrix is applied to symmetric unit 1, the second matrix to symmetric unit 2 and so on.
- 2 The matrices are defined as the transformation necessary to overlap the symmetric units onto the first symmetric unit. This means that the first matrix should be an identity matrix.
- 3 The file should not have any leading or trailing blank lines. Matrices should be separated by one and only one line. The matrix itself should be a 4x4 transformation matrix with one row per line and each column separated by one space only. For example, the identity matrix at the beginning of the file will look like the following:

```
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
```

Users can generate the 4x4 matrices for different transformations by using the matrix routines in VMD. Please refer to the matrix routine documentation in the VMD user guide for more details. The matrix file for the helical nitrilase system has been provided: helix-matrices.txt

5.4 Running the MDFF simulation

NAMD configuration files for MDFF can be generated similarly to the previous example. 2.

```
mdff setup -o symmetry -psf helix.psf
-pdb helix.pdb
-griddx helix-target.dx
-gridpdb helix-grid.pdb
-extrab {helix-extrabonds.txt helix-extrabonds-cispeptide.txt
helix-extrabonds-chirality.txt} -gscale 1.0 -minsteps 2000 -numsteps 500000
```

- 1 Now we need to edit the configuration file `symmetry-step1.namd` that we just created by adding in the parameters necessary for symmetry restraints. Open the file in any text editor and add the following lines anywhere:

```
symmetryRestrains on
symmetryfile helix-symmetry.pdb
symmetryk 200
symmetryMatrixFile helix-matrices.txt
symmetryfirststep 2001
symmetryfirstfullstep 502000
```

These parameters turn symmetry restraints on and let NAMD know where to look for the symmetry information. The `symmetryk` entry is a constant which scales the harmonic force applied by the restraint. This value is scaled down by the number of atoms in a symmetric unit. Users can define a per-atom force constant which assign force constant to individual atoms instead of to the whole symmetric unit, which is discussed in the NAMD user guide. One can vary the forces over time by modifying the force constant. In this case, the force constant will be linearly increased over time, allowing more conformational freedom at the beginning, working up to more rigid restraints as the molecule is fitted to the density. This is accomplished by the `symmetryfirstfullstep` entry which control when the force constant become the full assigned value (i.e. the `symmetryk` value). Setting this value to last timestep will linearly increase the force constant from the first timestep to the last timestep. More information about these parameters can be found in the symmetry restraint documentation in the NAMD User's Guide.

- 2 Quit VMD.
- 3 Run NAMD using the configuration files generated by VMD, i.e., run the following commands in a terminal (or submit them to a cluster):

```
namd2 symmetry-step1.namd > symmetry-step1.log
```

This step should take about 8 hours on a modern quad core desktop. If you don't want to wait, you can proceed to the analyzing section with the provided trajectory files.

5.5 Analyzing the results

The resulting trajectories will be saved to files `symmetry-step1.dcd`. If you want to continue working through the tutorial before the simulations are com-

plete, you can use the provided trajectory files `symmetry-step1-result.dcd` instead. First, you should load and view the trajectory file `no-symmetry-step1-result.dcd` provided. This trajectory is the result of running the above simulation with symmetry restraints turned off. You should notice that the last dimer is pulled away from the molecule, seen in Fig. 12. This is due to the extra density adjacent to these regions.

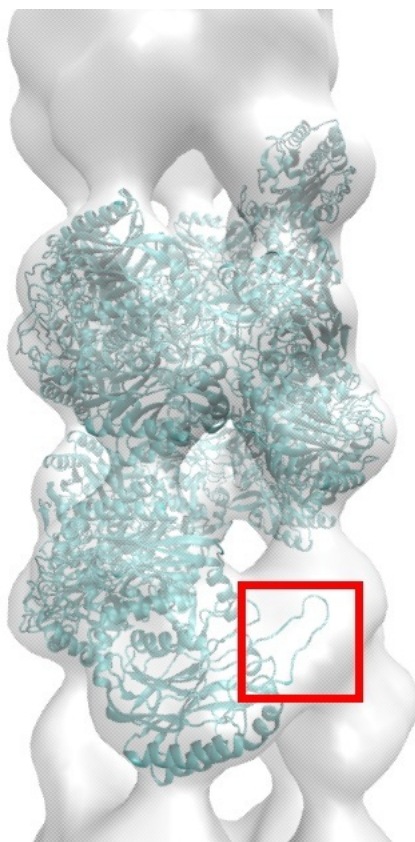


Figure 12: MDFF simulation of nitrilase without symmetry restraints. Red box indicates region of last dimer affected by adjacent density.

While it is possible to cut off extraneous portions of the map, this can introduce errors along the boundaries. Instead, we can use symmetry restraints to avoid these distortions. Load and view the trajectory obtained from your simulation using symmetry restraints `symmetry-step1.dcd` or use the trajectory file provided `symmetry-step1-result.dcd`. Compare the movements of the first and last dimers to those without symmetry restraints.

Acknowledgements

Development of this tutorial was supported by the National Institutes of Health (P41-RR005969 - Resource for Macromolecular Modeling and Bioinformatics).