Molecular Dynamics Flexible Fitting

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

March 28, 2019

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

Contents

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. Trabuco[1], Elizabeth Villa[1], Kakoli Mitra, Joachim Frank, and Klaus Schulten. Structure, 16:673-683, 2008.

The MDFF method can also be used to refine structures using low-resolution X-ray crystallography. This method, xMDFF was originally described in the manuscript:

xMDFF: Molecular dynamics flexible fitting of low-resolution X-Ray structures. Ryan McGreevy[1], Abhishek Singharoy[1], Qufei Li, Jingfen Zhang, Dong Xu, Eduardo Perozo, and Klaus Schulten. Acta Crystallographica D70, 2344-2355, 2014.

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

Molecular dynamics flexible fitting: A practical guide to combine cryo-electron microscopy and x-ray crystallography. Leonardo G. Trabuco[1], Elizabeth Villa[1], Eduard Schreiner, Christopher B. Harrison, and Klaus Schulten. Methods, 49:174-180, 2009.

Advances in the molecular dynamics flexible fitting method for cryo-EM modeling. Ryan McGreevy, Ivan Teo, Abhishek Singharoy, and Klaus Schulten. Methods, 100, 50-60, 2016.

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.

MDFF can also be used to refine structures from low-resolution X-ray crystallography through a method called xMDFF, covered as part of this tuto-

---

[1]Equal contribution

rial. xMDFF does not fit structures into cryo-EM maps, instead using crystallographic maps which must be generated using the PHENIX software suite. Because of this reliance on PHENIX, you will need to have a recent version installed and executable from the command line, which can be obtained from http://www.phenix-online.org/.

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. In Section 5, use of symmetry restraintes for MDFF of symmetric molecules is covered. In Section 6, xMDFF, an adaptation of MDFF for low-resolution X-ray crystallography is demonstrated. In Section 7, use of the MDFF Graphical User Interface (GUI) instead of the command line is demonstrated, along with interactive MDFF which allows users to manually manipulate their structure during an MDFF simulation.

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

Section 2: A simple MDFF example
`1ake-colores.pdb`
`1ake-initial.pdb`
`4ake-target.pdb`
`adk-step1-result.dcd`
`adk-step2-result.dcd`
Section 3: MDFF with explicit solvent
`adk-solvent-step1-result.dcd`
`adk-solvent-step2-result.dcd`
Section 4: MDFF with Domain Restraints
`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`
Section 5: MDFF with Symmetry Restraints
`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`

1 Overview of MDFF commands

At present, all MDFF commands in VMD are available 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
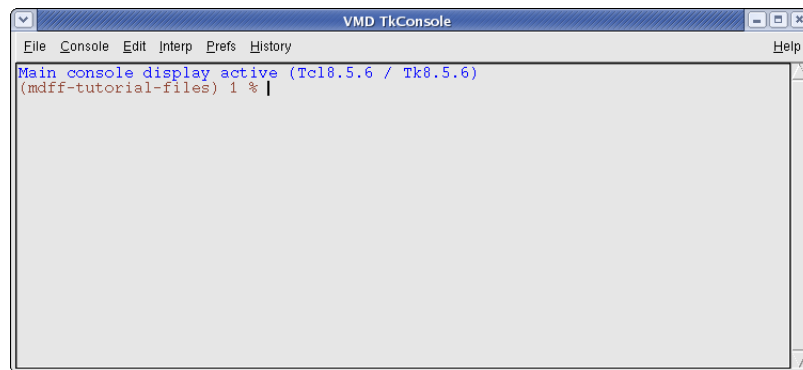start entering Tcl/Tk commands here.

Figure 1: The VMD Tk Console window.

2 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.

3 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
```

4 You can type any command to obtain its usage information. For exam-
ple, 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.

5 As of VMD version 1.9.2, the `mdff` plugin can also be used through a
graphical user interface, found in the "Modeling" section of the VMD
"Extensions" menu. This interface provides many of the same features
found in the command line version of the plugin in a more user friendly
package.

WARNING: *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 macro-molecule is flexibly fitted into a low-resolution density map of the same macro-molecule 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).
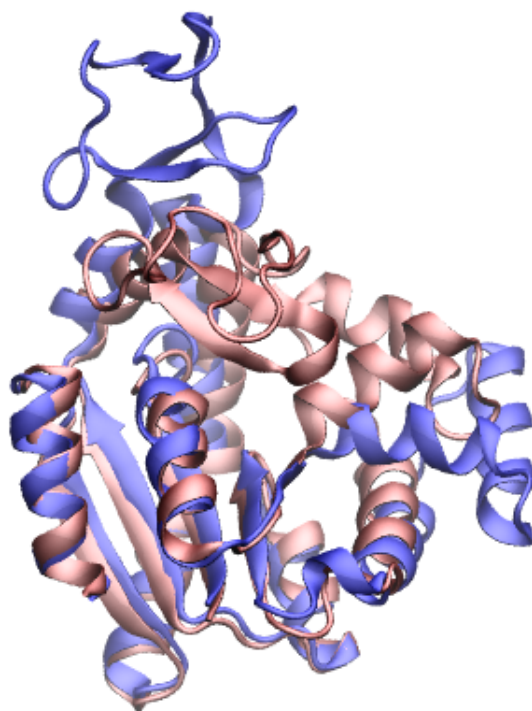
Figure 2: Initial and target structures shown in red and blue, respectively.

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`.

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).
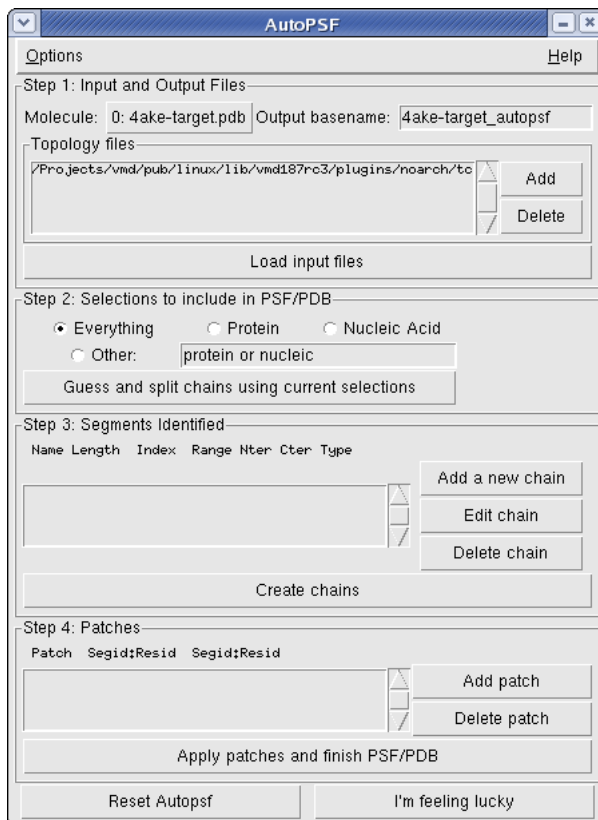
Figure 3: The VMD AutoPSF window.

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 dx file format, type:

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

7 Load the simulated map in VMD by typing

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

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

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

and

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

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 2.4 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.
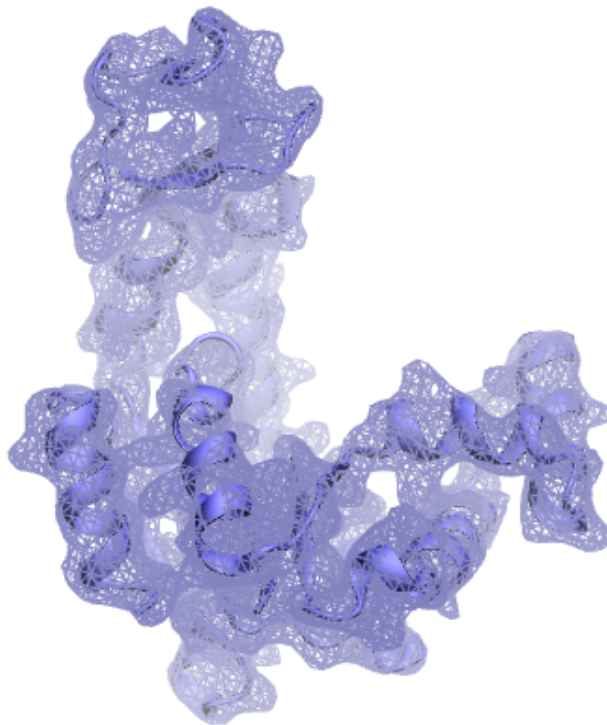
Figure 4: Target structure and simulated map shown in cartoon and mesh, respectively.

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.dx -o
   4ake-target_autopsf-grid.dx
```

2.3   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 `voltool fit` command in VMD.

1 Run the following commands in VMD's TkConsole:

```
mol new 1ake-initial.pdb
set sel [atomselect top "protein and noh"]
voltool fit $sel -res 5 -i 4ake-target_autopsf.dx
```

2 After the rigid body fitting is complete, the selected atoms should be docked inside the density map. To save this structure, type the following command: `$sel writepdb 1ake-initial-docked.pdb`

2.4 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 docked structure in VMD by typing:

```
mol new 1ake-initial-docked.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 the choose the correct molecule in the AutoPSF plugin. You should be able to generate the files `1ake-initial-docked_autopsf.psf` and `1ake-initial-docked_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 1ake-initial-docked_autopsf.psf
  -pdb 1ake-initial-docked_autopsf.pdb -o 1ake-initial-docked_autopsf-grid.pdb
```

2.5 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 Type `ssrestraints` to check the usage information.

2 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 1ake-initial_autopsf.psf
  -pdb 1ake-initial-docked_autopsf.pdb -o 1ake-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.

3 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 1ake-initial-docked_autopsf.psf
mol addfile 1ake-initial-docked_autopsf.pdb
```

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

```
cispeptide restrain -o 1ake-extrabonds-cispeptide.txt
```

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

```
chirality restrain -o 1ake-extrabonds-chirality.txt
```

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 1ake-initial-docked_autopsf.psf
  -pdb 1ake-initial-docked_autopsf.pdb
  -griddx 4ake-target_autopsf-grid.dx
  -gridpdb 1ake-initial-docked_autopsf-grid.pdb
  -extrab {1ake-extrabonds.txt 1ake-extrabonds-cispeptide.txt
  1ake-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 1ake-initial-docked_autopsf.psf
   -pdb 1ake-initial-docked_autopsf.pdb
   -griddx 4ake-target_autopsf-grid.dx
   -gridpdb 1ake-initial_autopsf-grid.pdb
   -extrab {1ake-extrabonds.txt 1ake-extrabonds-cispeptide.txt
   1ake-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. You can also improve the speed of the simulation by using more processors by passing namd2 the +p$ option, where $ is the number of processors you wish to use.

REMARK: The optimal choice of the scaling factor ξ, i.e. the gscale parameter, depends on the system to be fitted and the map. The higher the value, the stronger the forces will be acted on the system to fit the map. In general a gscale of 0.3 works fine for step 1 but for step 2, the minimization step, it is advised to test a few different gscale, such as 1, 5 and 10 and choose the one that works best for your map and system.

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-docked_autopsf.psf
mol addfile 1ake-initial-docked_autopsf.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.
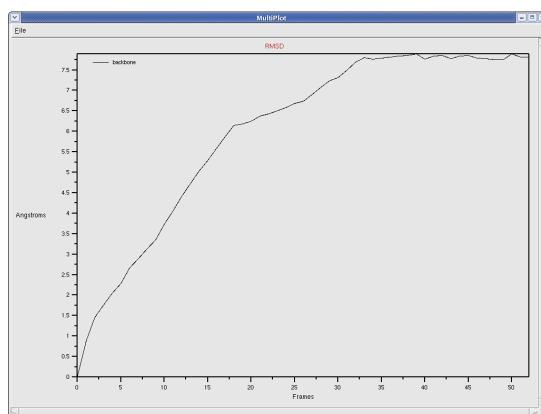
Figure 5: Backbone RMSD with respect to the initial structure for the MDFF simulation.

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.
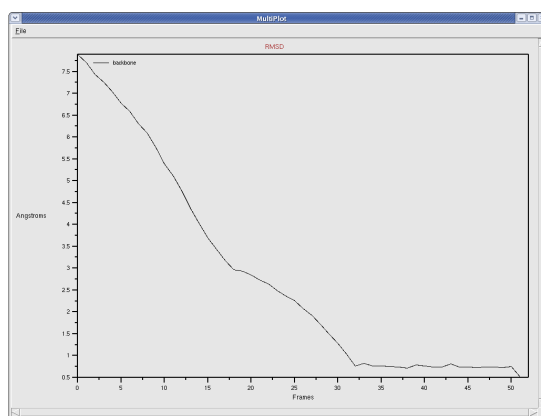

Figure 6: Backbone RMSD with respect to the target structure for the MDFF simulation.

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
```

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.dx -res 5
```
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.

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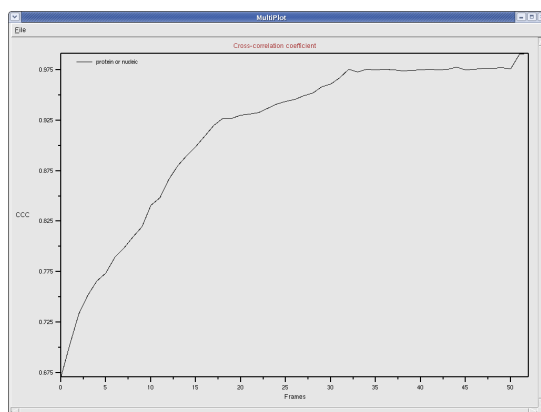
Figure 7: Cross-correlation coefficient between target density map and each frame of the MDFF simulation.

4 Calculate the initial CCC:

```
$selall frame 0
mdff ccc $selall -i 4ake-target_autopsf.dx -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.dx -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 `1ake-initial-docked_autopsf.psf` and `1ake-initial-docked_autopsf.pdb`.

1 Start a new VMD session.

2 Load the initial structure you prepared for MDFF in vacuo in the previous section:

```
mol new 1ake-initial-docked_autopsf.psf
mol addfile 1ake-initial-docked_autopsf.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. We would like the target map to fall 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).
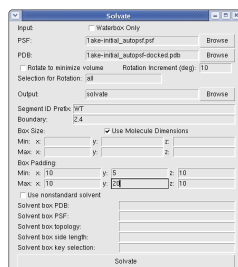
Figure 8: The VMD Solvate window.

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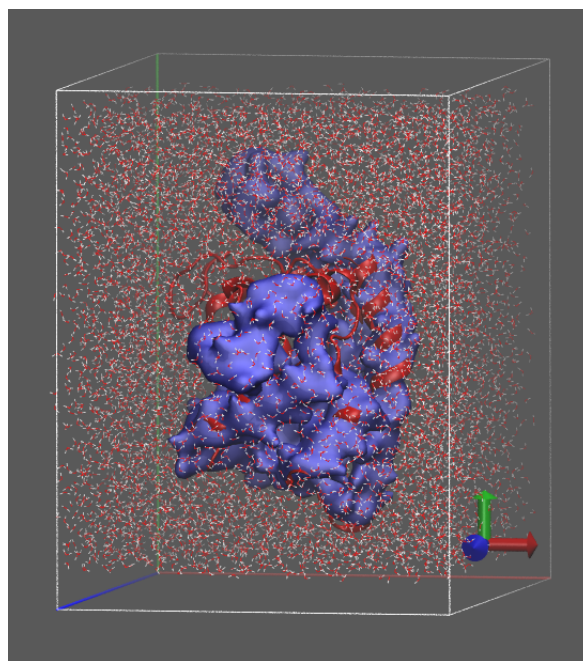
Figure 9: By loading the target density map you can visually ensure that the target density falls within the boundaries of the water box. If part of the target density is outside the water box, you should adjust the padding accordingly and regenerate the water box.

plugin. In the VMD Main Window, choos 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
```

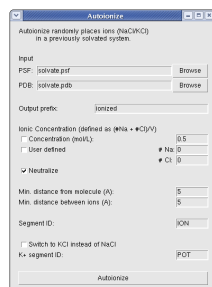**7** Generate restraints to prevent cis/trans peptide transitions and chirality errors:

```
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). An option, `gridforcechecksize off` or `mgridforcechecksize 0 off` if using `mgridforce`, can be used to ignore the warning. Additionally, to address this issue directly, we can trim the target map so that it lies completely within the water box.

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

```
voltool trim -i 4ake-target_autopsf-grid.dx -amt 7 7 7 7
  -o 4ake-target_autopsf-grid-trimmed.dx
```

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 ad CCC. Did the use of explicit solvent improve the MDFF 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.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. You will also need to create a psf file following the same steps outlined there using the `acoasyn-initial.pdb` model.

1 Start a new VMD session.

2 Load the initial structure in VMD by typing:

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

3 Use 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 the choose the correct molecule in the AutoPSF plugin. Follow the same steps as before to make acoasyn-initial_autopsf.psf and acoasyn-initial_autopsf.pdb.

4 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_autopsf.psf -pdb acoasyn-initial_autopsf.pdb
  -o acoasyn-grid.pdb
```

5 Generate secondary structure restraints as in the previous section:

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

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

```
mol new acoasyn-initial_autopsf.psf
mol addfile acoasyn-initial_autopsf.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 conaining 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_autopsf.psf
mol addfile acoasyn-initial_autopsf.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_autopsf.psf
  -pdb acoasyn-initial_autopsf.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
```
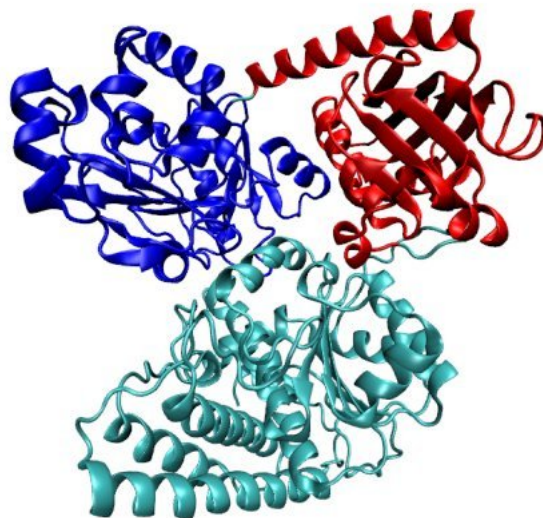
Figure 11: The two domains defined by the pdb file. Domain 1 in red and domain 2 in blue.

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 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.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 Load the initial structure in VMD by typing:

```
mol new helix.pdb
```

3 Use 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 the choose the correct molecule in the AutoPSF plugin. Follow the same steps as before to make helix_autopsf.psf and helix_autopsf.pdb.

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

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

5 Generate secondary structure restraints as in previous section:

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

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

```
mol new helix_autopsf.psf
mol addfile helix_autopsf.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_autopsf.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 `measure fit` command in VMD. Please refer to the 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_autopsf.psf
  -pdb helix_autopsf.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 before the `source mdff_template.namd` line:

```
symmetryRestraints 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. Thie 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 complete, 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.

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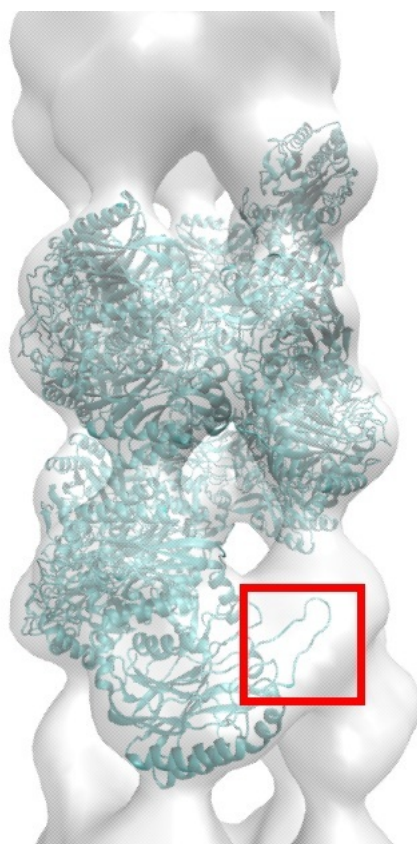
Figure 12: MDFF simulation of nitrilase without symmetry restraints. Red box indicates region of last dimer affected by adjacent density.

6   xMDFF: MDFF for Low-Resolution X-ray Crystallography

Although originally developed for fitting crystal structures into cryo-EM densities, MDFF can also be used to refine structures from low-resolution x-ray crystallographic diffraction data, termed xMDFF. For use with low-resolution X-ray crystallography, the MDFF protocol is modified to work with densities derived from molecular replacement, which uses the phases ϕ_{calc} calculated from a tentative model and the amplitudes $|F_{obs}|$ from the X-ray diffraction data. The density is biased by the model, but contains enough information from the $|F_{obs}|$ to determine the experimental structure. These densities are created using the PHENIX software suite to generate $2mF_{obs} - DF_{calc}$ maps. Because of this reliance on PHENIX, you will need to have a recent version installed and executable from the command line, which can be obtained from http://www.phenix-online.org/.

Unlike in standard MDFF simulations, the density map in xMDFF changes throughout the course of the refinement. Once the tentative model is fit to the generated density, the xMDFF-fitted structure provides new ϕ_{calc} that, together with $|F_{obs}|$, are used to regenerate the electron density. The fitted structure is then employed as an updated search model to be driven into the new density map obtained from a molecular replacement procedure, and this process continues iteratively until a sufficiently low R_{free} and R_{work} are obtained.

Please note that you should at least briefly read through Section 2 or refer back to it as you work through this section of the tutorial. xMDFF shares many of the same initial steps with MDFF which are covered in earlier sections. Overlapping material will be presented here in a complete yet more concise form to avoid excessive repetition.

6.1   Preparing the initial structure

Normally xMDFF would require the low-resolution experimental reflection data and an initial homology model or predicted structure to begin refinement. For the following example, we will use the open conformation of the D ribose binding protein (PDB: 1URP) as an initial model and refine it against synthetically created diffraction data from a known closed conformation (PDB: 2DRI) at 5 Å resolution (Fig. 13).

The initial steps of xMDFF are identical to a standard MDFF simulation outlined in Section 2 which you should refer to for further information. First we must generate a PSF file to provide NAMD with connectivity and partial aotmic charge information.

1 Load the initial structure in VMD by typing:

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

2 Use 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
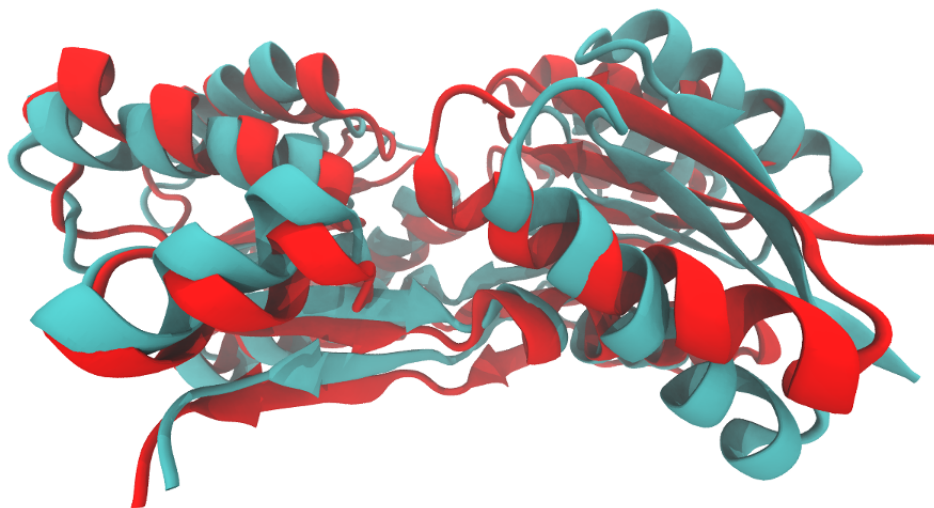
Figure 13: The open conformation of the ribose binding protein (PDB: 1URP) colored in cyan will be used as an initial phasing model to be refined against relfection data from the closed conformation (PDB: 2DRI) colored in red.

Reset AutoPSF button and the choose the correct molecule in the AutoPSF plugin. You should be able to generate the files 1urp-initial_autopsf.psf and 1urp-initial_autopsf.pdb.

As in previous sections, we need to generate secondary structure, chirality, and cispeptide restraints to help prevent overfitting during the simulation.

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 1urp-initial_autopsf.psf
  -pdb 1urp-initial_autopsf.pdb -o 1urp-extrabonds.txt -hbonds
```

4 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 1urp-initial_autopsf.psf
mol addfile 1urp-initial_autopsf.pdb
```

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

```
cispeptide restrain -o 1urp-extrabonds-cispeptide.txt
```

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

```
chirality restrain -o 1urp-extrabonds-chirality.txt
```

7 You now need to generate a PDB file containing the per-atom scaling
factors w_j in Equation 1. These scaling factors are set to the atomic
mass by the `mdff gridpdb` command in the VMD `mdff` plugin, which you
can load by typing `package require mdff` in the VMD Tk Console. In
xMDFF it is especially useful to only couple certain atoms at different
stages of refinement. For example, the initial density maps are usually
so poor and noisy that you may want to only designate the α-carbon
or backbone for having xMDFF forces applied to them. This allows for
greater flexibility in the structure for increased sampling early on and
to avoid over-fitting to any noise in the potential. As the refinement
progresses, you can couple more atoms (e.g. side chains) to improve the
overall fit. As such, you should generate three PDB files here which will
select α-carbons, backbone and all protein atoms excluding hydrogen.

```
mdff gridpdb -psf 1urp-initial_autopsf.psf
  -pdb 1urp-iniitial_autopsf.pdb -o cagrid.pdb -seltext "name CA"

mdff gridpdb -psf 1urp-initial_autopsf.psf
  -pdb 1urp-iniitial_autopsf.pdb -o backgrid.pdb -seltext "backbone"

mdff gridpdb -psf 1urp-initial_autopsf.psf
  -pdb 1urp-iniitial_autopsf.pdb -o nohgrid.pdb -seltext "protein and noh"
```

6.2   Setting Up NAMD Configuration Files for xMDFF

In this section we will create the NAMD configuration files and tcl script required
to run xMDFF.

1 You must first generate a NAMD configuration file, which 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 2dri -psf 1urp-initial_autopsf.psf
  -pdb 1urp-initial_autopsf.pdb
  -griddx step1.dx
  -gridpdb cagrid.pdb
  -extrab {1urp-extrabonds.txt 1urp-extrabonds-cispeptide.txt
  1urp-extrabonds-chirality.txt} -gscale 0.1 -numsteps 700000
  ---xmdff  -refs 2DRI.mtz
```

Most options above specify the names of files you generated in previous
steps and should be self-explanatory. As discussed previously, we will be

initially applying forces only to the α-carbons using "cagrid.pdb". The option `-gscale` defines the scaling factor ξ in Equation 1. In addition to the initial α-carbon coupling, we will use a low scaling factor (gscale 0.1) to decrease the magnitutde of the xMDFF forces and allow for greater initial flexibility. Unlike in previous sections, the potential map specified by `-griddx` does not yet exist, but instead will be generated automatically during the simulation with the name given. The flag `---xmdff` causes the plugin to accept xMDFF specific options as well as generate a few additional lines in the NAMD configuration file, which will be discussed next. The `-refs` option is required for any xMDFF simulation, and refers to the file containing the reflection data you wish to refine the structure against. This file can be in either the mtz or cif format. Additional options we are not specifying here will also be discussed next.

3 Open the configuration file 2dri-step1.namd with a text editor. We will be taking a look at some of the options in this file along with xMDFF specific entries that differ from normal MDFF simulations. The first things you should notice near the beginning of the file are the variables:

```
set REFINESTEP 20000
set REFS 2DRI.mtz
set BFS 0
set MASK 0
set CRYSTPDB 0
```

`REFINESTEP` sets the number of timesteps in between regenerating the density map from the new phases taken from the current structure. 20,000 was found to provide enough time for the structure to fit to the current density. You can increase this number to allow for increased sampling of each density, or lower it to capture smaller changes in phases, at the cost of computational speed. 20,000 is the defualt value, but this can be changed with the `-refsteps` option of `mdff setup`.

`REFS` indicates the reflection data file as discussed previously.

`BFS` turns on (1) or off (0) individual adp refinement using phenix for the calculation of B-factors. This option causes the refinement to take longer every time a map is regenerated, and is not necessarily required for this example so we will leave it turned off. This option can be turned on with `mdff setup` using the `---bfs` flag.

`MASK` turns on (1) or off (0) masking the generated density so that only the density immediately around the structure is used during fitting. Masking the density can help remove noise, but is not always needed. This option can be turned on with `mdff setup` using the `---mask` flag.

`CRYSTPDB` gives the name of a text file (can be a PDB) with a properly formatted CRYST1 line with space group and unit cell information. This is not always required as the reflection data file usually contains this information, but in the case that it does not or is improperly specified, you can supply the information with this file. This file can be given with `mdff setup` using the `-crystpdb` option.

4 Now open the file xmdff_template.namd with a text editor. This files contains more NAMD configuration parameters, including xMDFF specific commands. First take a look at the following command near the bottom of the file: `if {[info exists INPUTNAME]}`

```
set env(VMDARG) [list $PSFFILE $INPUTNAME $GRIDFILE $REFS $BFS
$MASK $CRYSTPDB $MASKRES $MASKCUTOFF $XMDFFSEL $AVERAGE]
exec -ignorestderr vmd -dispdev text -e xmdff_phenix.tcl > map.log
} else {
set env(VMDARG) [list $PSFFILE $PDBFILE $GRIDFILE $REFS $BFS $MASK
$CRYSTPDB $MASKRES $MASKCUTOFF $XMDFFSEL $AVERAGE]
exec -ignorestderr vmd -dispdev text -e xmdff_phenix.tcl > map.log
```

This section marks the beginning of the most significant changes to the standard MDFF NAMD configuration file. This first block generates the initial density map to which the structure will be fit. If you are restarting from a previous run, it will use the last restart coordinates provided by INPUTNAME as the phasing model. If you are beginning from just a PDB, then that will be used instead. The important part of this section of code is the line beginning `exec -ignorestderr vmd ...` which runs a tcl script through VMD to perform the required steps of map generation. This script will be investigated in the next section.

```
if {$ITEMP != $FTEMP} {
set ANNEALSTEP [expr abs($FTEMP-$ITEMP)*100]
run $ANNEALSTEP
set env(VMDARG) [list $PSFFILE $OUTPUTNAME $GRIDFILE $REFS $BFS
$MASK $CRYSTPDB $MASKRES $MASKCUTOFF $XMDFFSEL $AVERAGE]
exec -ignorestderr vmd -dispdev text -e xmdff_phenix.tcl > map.log

for {set i 0} {$i < [llength $REFS]} {incr i} {
set grid [expr [llength $GRIDFILE]-[llength $REFS]+$i]
reloadGridforceGrid $grid
}
}
```

This block of code will perform perform the appropriate amount of steps for any simulated annealing you might do, followed by generating the

map just as in the previous section. There is a new command here, **reloadGridforceGrid**, which tells NAMD to reload the potential map derived from the density which we are fitting to (since we just created a new one).

```
for {set i 0} {$i < $TS/$REFINESTEP} {incr i} {
run $REFINESTEP
set env(VMDARG) [list $PSFFILE $OUTPUTNAME $GRIDFILE $REFS $BFS
$MASK $CRYSTPDB $MASKRES $MASKCUTOFF $XMDFFSEL $AVERAGE]
exec -ignorestderr vmd -dispdev text -e xmdff_phenix.tcl > map.log
for {set j 0} {$j < [llength $REFS]} {incr j} {
set grid [expr [llength $GRIDFILE]-[llength $REFS]+$j]
reloadGridforceGrid $grid
}
}
```

This is where the standard refinement occurs by running NAMD for the number of specified refinement steps while iteratively re-generating the map each time from the latest restart coordinates.

5 Using a text editor, open xmdff_phenix.tcl. This file is a tcl script which is run using VMD to prepare a structure file for map generation, as well as calling PHENIX. There is nothing in this file you should need to modify, however since it is just making use of VMD's scripting capabilities with tcl, you are able to make a wide variety of changes to suit your particular system.

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

```
namd2 2dri-step1.namd > 2dri-step1.log
```
This will begin the NAMD simulation and refinement process. If you do not wish to wait for this simulation to complete, you may use the supplied 2dri-step1_result.dcd file and skip to the analysis in the next section.

6.3   Analysis of xMDFF refinements

1 To analyze the results from the first step of the xMDFF refinement, you will need to start VMD and load your structure files and trajectory:
```
mol new 1urp-initial_autopsf.psf
mol addfile 1urp-initial_autopsf.pdb
mol addfile 2dri-step1.dcd waitfor all
```
or if you are using the supplied dcd:
```
mol addfile 2dri-step1_result.dcd waitfor all
```

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. 14 should appear. Note how the RMSD levels off toward the end of the simulation.
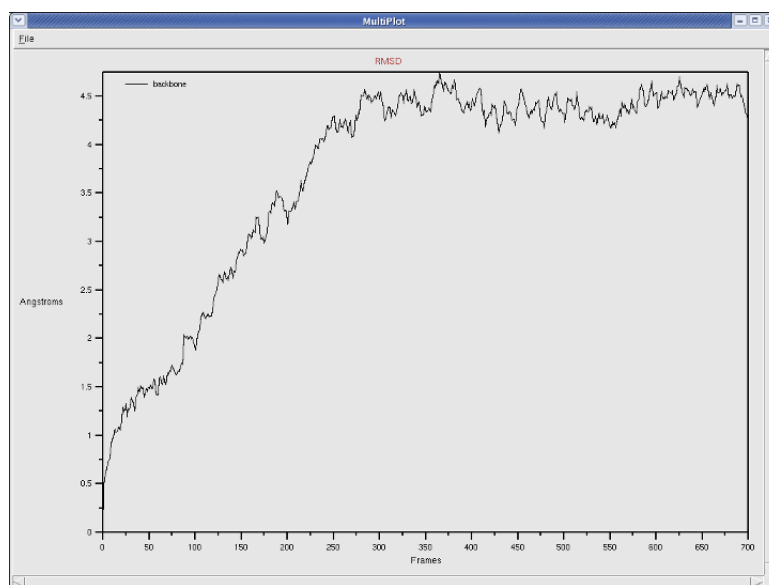

Figure 14: RMSD plot of refinement, showing this step has converged.

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

```
mdff check -rmsd -refpdb 2DRI.pdb
```
A window similar to the one depicted in Fig. 15 should now appear. Note how the RMSD decreases (the fitting improves) which indicates the refinement of the structure over the course of the simulation.

4 We will also analyze the refined structure using a PHENIX program, `phenix.model_vs_data` which will provide R_{work} and R_{free} values in addition to structural statistics from Molprobity (e.g. % Ramachandran Favored Angles). First we should obtain a PDB of the final frame and refine the beta factors:

In VMD make sure the molecule containing your refinement trajectory is the "top" molecule, then in the VMD TKConsole type:
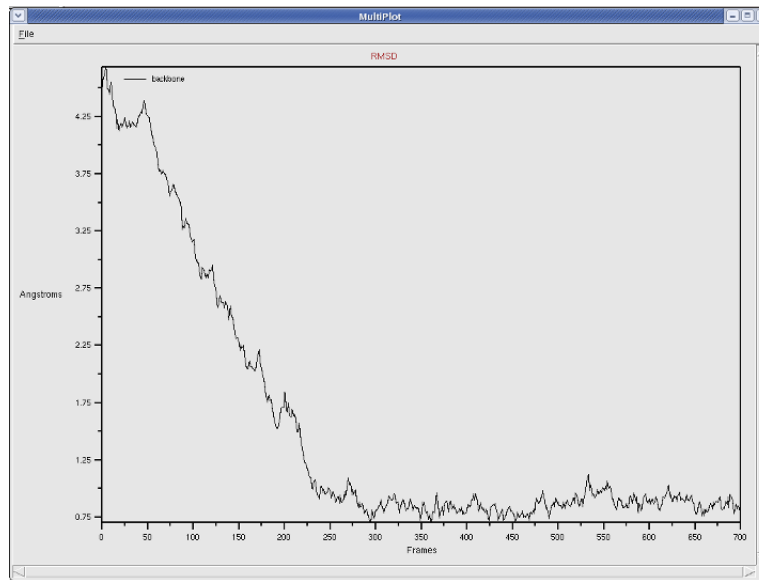
Figure 15: RMSD plot of refinement relative to target structure, indicating refinement.

```
set sel [atomselect top "protein and noh"]
$sel frame last
$sel writepdb final.pdb
```
Open final.pdb with a text editor and delete the first line beginning with "CRYST1". Additionally, you will have to change every occurrence of "HSD" in the file with "HIS". Any text editor with a 'search and replace' feature can do this, for example using vi you can type: %s/HSD/HIS/g". Then, run phenix from a normal shell terminal:

```
phenix.refine final.pdb 2DRI.mtz refinement.refine.strategy=individual_adp
---overwrite
```

Now we will run `phenix.model_vs_data` on the structure with the beta factors:
```
phenix.model_vs_data final_refine_001.pdb 2DRI.mtz > model.log
```

Once this is complete, you can open model.log with a text editor and view the statistics of your refined structure.

5 One thing you may notice from the RMSD plots and `phenix.model_vs_data` output, is that our structure isn't as refined as it could be. To do this, you will have to run further simulations using the previous one as input

while coupling additional atoms to the density, increasing the global scaling factors, and using any additional techniques such as kicked maps and beta factor sharpening. It is also often useful to bring the simulation temperature down to 0K for the final structure by setting the `FTEMP` variable to 0 in the last NAMD simulation. Some example steps to set up the next simulation are:

Copy 2dri-step1.namd to 2dri-step2.namd, then open 2dri-step2.namd with a text editor and make the following changes:

Edit the following lines to reflect these new values:
`set GRIDPDB backgrid.pdb` to couple all backbone atoms to the density.
`set GSCALE 0.3` to increase the global scaling factor of the applied forces.

Add the following line just before `OUTPUTNAME`:
`set INPUTNAME test-step1` to restart from the previous step's progress.

Edit the following line:
`set OUTPUTNAME test-step2` to name the output of this simulation as step2.

Run NAMD again using the new configuration file:
`namd2 2dri-step2.namd > 2dri-step2.log`
Once this simulation is complete, repeat the previous steps by making a 2dri-step3.namd and using the nohgrid.pdb as the `GRIDPDB` and increasing the `GSCALE` to 0.6. You should also experiment with using kicked maps and beta factor sharpening for different steps and analyze the difference in the resulting structures. An example of a more complete refinement trajectory, 2dri_result.dcd, can be found in the tutorial files.

7 MDFF GUI and Timeline Analysis

As of VMD version 1.9.2, the `mdff` plugin can also be used through a graphical user interface, found in the "Modeling" section of the VMD "Extensions" menu. This interface provides many of the same features found in the command line version of the plugin in a more user friendly package. The GUI can be used to easily set up both MDFF and xMDFF simulations, as well as launch, connect to, and analyze interactive MDFF simulations. Additionally, the Timeline plugin in the "Analysis" section of the VMD "Extensions" menu now contains functionality for calculating, displaying, and analyzing the local cross correlation of a structure over the entire fitting trajectory. The following sections will present the steps for setting up and analyzing the same system found in Section 2, but now using the MDFF GUI and Timeline analysis. If you are learning how to

use MDFF for the first time, it is best to start with Section 2 before returning to this section.

7.1   Setting up simulations with the MDFF GUI

Before proceeding through this section and setting up your MDFF simulation, make sure you have worked through Section 2. You will need the simulated density, psf, and pdb from Section 2.1 to proceed. If you have these files, you can begin by opening up the MDFF GUI from the "Modeling" section of the VMD "Extensions" menu. Please note you may need to type the `package require mdff` command into the TkConsole before launching the GUI.
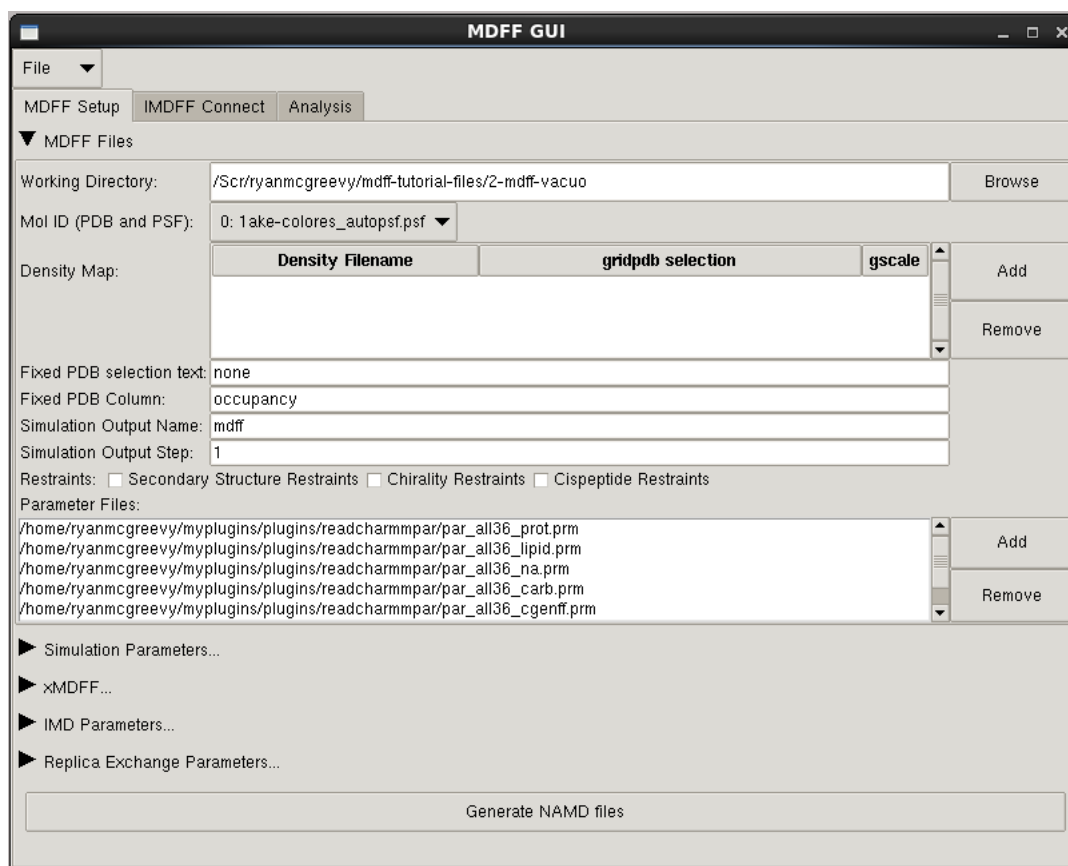

Figure 16: MDFF Files tab of the MDFF GUI, where parameters for the files required by MDFF are set

    1 Upon starting the MDFF GUI, you will see several drop-down sections relating to various aspects of setting up the MDFF simulation. Begin by

opening the 'MDFF Files...' section, seen in Fig. 16. Here you will choose
the already loaded molecule containing both the psf and pdb files obtained
from Section 2, `1ake-initial_autopsf.psf` and `1ake-initial_autopsf-docked.pdb`.
The 'Working Directory' indicates where all of the output files will be saved
when we generate them later.

2 After selecting the molecule, you will have to select the density map
`4ake-target_autopsf.situs` which you previously generated in Section
2.1, by clicking the 'Add' button. A window will open where you can
browse for the map. Additionally, the window contains an input box for
selection text for the 'gridpdb' file which marks the atoms being affected
by the density-derived potential. You may leave the default text for this
tutorial. Similarly, you can also set the grid scaling factor , which again
you can leave as the default value. When you are satisfied with the set-
tings, click the 'Add Density' button.

3 In the 'Simulation Output Name' field, you can select a name for your
MDFF simulation. This can remain the default for this tutorial, as well
as the 'Simulation Output Step'. It is important to note that if you place
any number greater than 1 in the Step field, the generated NAMD files will
automatically load the output from the previous simulation step which will
act as a starting point for your next simulation. This can be very useful
for quickly setting up multi-step fitting workflows.

4 As further discussed in Section 2.4, restraints are an important aspect
of MDFF simulatiuns to prevent overfitting. You should select all three
restraint check boxes so that the corresponding files will be created.

5 Next, you should close the 'MDFF Files' subsection of the GUI and open
the 'Simulation Parameters' subsection, as seen in Fig. 17. This is the
location of basic NAMD simulation parameters, such as temperature and
time steps. All of the default values can remain for this tutorial. You
may want to check the box labeled 'gridforcelite', which will tell NAMD
to use a faster, but slightly less accurate, interpolation for calculating the
forces derived from the density map. In this section you can also select
the environment of your MDFF simulation, e.g. vacuum, explicit solvent
with periodic boundary conditions, or a generalized Born implicit solvent.
Vacuum should be used for this tutorial.

The next subsection of the GUI, 'xMDFF...' will not be used here, but
anyone interested in applying MDFF to low-resolution crystallography should
begin by reading Section 6. You can use the files from that section to set up an
xMDFF simulation similar to what you have done here, but with the addition
of setting the parameters in the xMDFF GUI subsection.

You now have all of the options set to be able to create all the files you need
for the MDFF simulation. If you wish to run the MDFF simulation and move
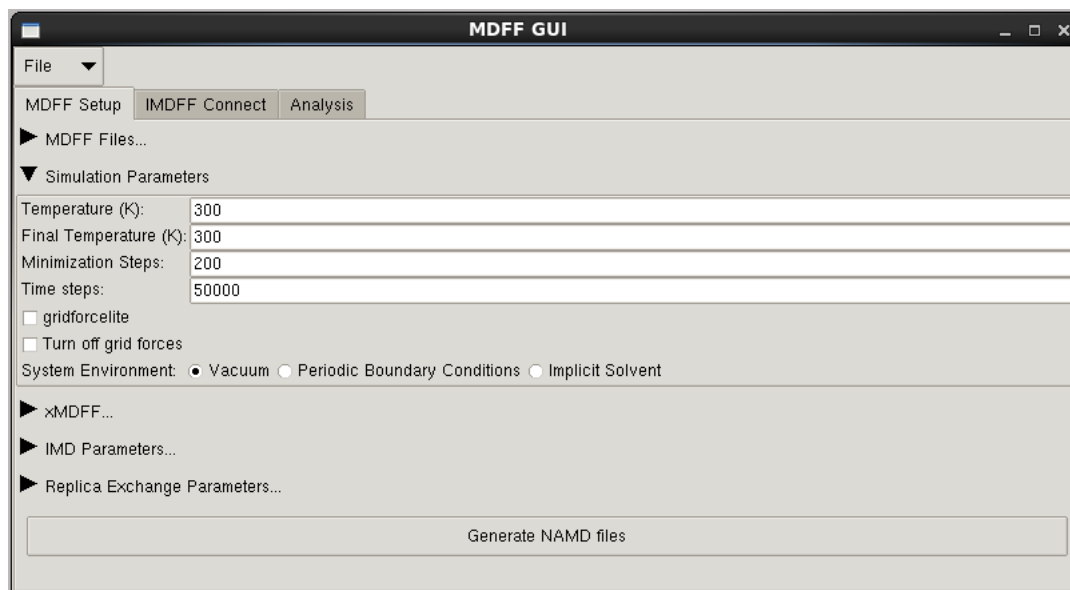
Figure 17: Simulation Parameters subsection of the MDFF GUI.

on to analyzing the trajectory, you can simply select the 'Generate NAMD files' button at the bottom of the 'MDFF Setup' tab of the MDFF GUI. However, if you wish to run an interactive MDFF simulation, please skip the following commands and proceed to Section 7.2. Before continuing either way, you should first save your current settings, using the 'File' drop down menu at the top of the GUI. You can select 'Save Settings...' to write the current GUI settings to a tcl file. This file can later be loaded using 'Load Settings...', which will set all of your saved parameters and load any structure and density files you had previously loaded. If you do choose to generate the NAMD files now, this button will automatically generate all the files you need to begin your simulation. Once this process is complete, you can start your simulation by running the following command in a terminal: `namd2 mdff-step1.namd > mdff-step1.log`

This step should take about 35 minutes on a single processor. If you don't want to wait, you can proceed to the Timeline section, 7.3 and use the provided trajectory files. You can also improve the speed of the simulation by using more processors by passing namd2 the +p$ option, where $ is the number of processors you wish to use.

7.2   Interactive MDFF

Interactive molecular dynamics (IMD) allows you to view and manipulate a NAMD simulation in real-time using VMD. Additional information on IMD can be found here. IMD allows users to selectively apply forces to your struc-

ture which can be very useful during MDFF to help guide the structure into the regions of density which you believe it should fit. Complex structures, poorly resolved maps, or starting structures requiring very large or complex conformational changes to fit into the density, may necessitate such manual intervention. Sometimes pieces of a structure can even become trapped in local density minima which they do not belong in, and manually dragging them out can improve the effeciency and end result of your MDFF fitting. This particular test case, 1AKE, does not have any such issues, however you may still want to experiment to see how IMD works.

With the MDFF GUI, we can now easily set up, launch, connect to, and even analyze interactive MDFF (IMDFF) simulations. The 'IMD Parameters...' subsection of the MDFF GUI on the MDFF Setup tab contains options related to setting up IMDFF simulations.

1 Open the 'IMD Parameters' subsection of the MDFF GUI and check the 'IMD' box to turn on IMDFF. Next, check the 'Wait for IMD Connection' box so NAMD will wait until you have connected VMD to the simulation before beginning. 'Ignore IMD Forces' tells NAMD to ignore any steering forces that could be applied through VMD (e.g. use this setting if you only wish to observe your simulation). Do NOT check this box for this tutorial. The IMD Port, Frequency, and Keep Frames settings can all be kept as default. The IMD Port is the computer network port over which VMD and NAMD communicate. IMD Frequency sets the frequency of NAMD simulation steps sent from NAMD to VMD. IMD Keep Frames sets how many of the steps NAMD sends to VMD are saved in VMD. The simulation steps will still be saved to a .dcd regardless, but you may want to save them in VMD immediately as they come in. Be aware that this may use a significant amount of computer memory.

2 The next settings, IMD Server and Processors, are related to where and how the NAMD simulations will be run. By default, the simulations are run on your local machine by calling the `namd2` command with the selected number of processors. This assumes that `namd2` is on your computer's PATH. If you wish to adjust how NAMD is called or add additional server definitions, you can add them to a MDFF GUI settings file. To see how the default server is set, save your settings file (discussed at the end of the previous section) and open the file with a text editor. You should see near the bottom something like:

```
MDFFGUI::gui::add_server "Local" {
maxprocs 12
namdbin {namd2 +p%d}
jobtype local
timeout 20
numprocs 12
}
```

Here you can modify the "Local" server, or create a new entry and name it

something unique (i.e. not "Local"). You can change how `namd2` is called in the `namdbin` setting, the default number of processors to use, and the timeout (in seconds) for how long VMD should wait after attempting to contact NAMD before giving up.

3 With all of the parameters now set, you should click the 'Generate NAMD files' button at the bottom of the MDFF Setup tab, seen in Fig. 18. This button will automatically generate all the files you need to begin your simulation and save them in your selected working directory.
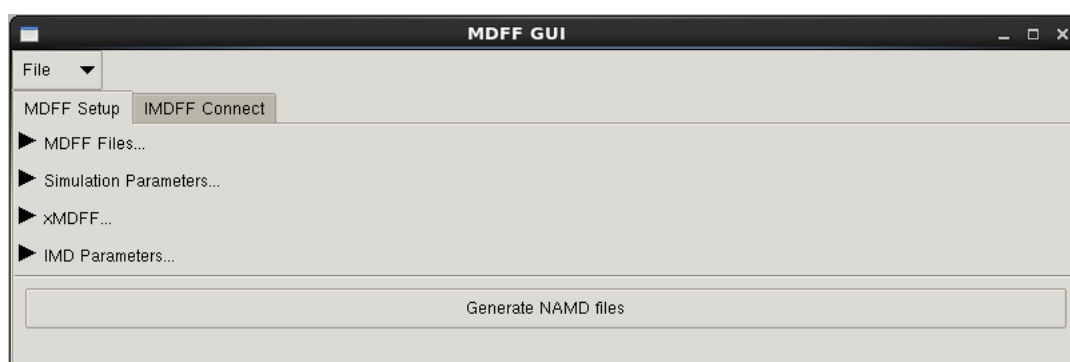

Figure 18: The Generate NAMD files button creates all of the files needed for MDFF and saves them in your working directory

4 Navigate to the 'IMDFF Connect' tab of the MDFF GUI, shown in Fig. 19. This section of the GUI contains buttons for submitting, connecting to, pausing, stopping, and analyzing the IMDFF simulation set up in the MDFF Setup tab.

Figure 19: IMDFF Connect tab of the MDFF GUI, where interactive MDFF simulations can be submitted, connected to, paused, terminated, and analyzed

5 Before (or after) connecting to a running IMDFF simulation, you can open the 'Cross Correlation Analysis' subsection in the IMDFF Connect tab to perform real-time analysis of your fitting. To do so, check the 'Calculate Real-Time Cross Correlation' box. You can also set the atom selection for the piece of the structure you wish to analyze, as well as the resolution of the map (the defaults for both are fine for this tutorial). You can also set a threshold value, where any density values less than the threshold will not be included in the analysis. More information on choosing a threshold is given in Section 7.3. The cross correlation value for each simulation frame will be graphed in the area at the bottom of this tab, as seen in Fig. 20. This is a convenient way of determining how well your fitting is progressing and whether the simulation has converged.

6 You should now start your simulation by selecting the "Submit" button. Wait several seconds after clicking the 'Submit' button, then select the 'Connect' button. VMD will make several attempts (dictated by the amount of time in the selected server's timeout parameter) to connect to the running NAMD simulation.

7 Once VMD is connected to the simulation, you should see your structure moving in the VMD display window. This VMD window is showing your running MDFF simulation in real-time. You can pause or terminate the simulation at any time with the "Pause" and "Finish" buttons in the IMDFF Connect tab of the MDFF GUI.

8 When connected to a running IMDFF simulation, you can not only view the progress of your fitting, but manually manipulate the structure as well. To do so, in the Mouse menu of the VMD Main window, select "Mouse -> Force -> Atom". You may alternately select "Residue", or "Fragment" to apply forces to whole residues or fragments. Your mouse can now be used to apply forces to your simulation. Click on an atom, residue, or fragment and drag to apply a force. Click quickly without moving the mouse to turn the force off. The structure you are using for this tutorial, 1AKE, has no trouble fitting into the density and does not require interactive MDFF, but you should take this oppurtunity to try out this feature and drag pieces of the structure around. Please note however that the simulation will end once the previously set number of steps have been calculated. You can always change the number of time steps, regenerate the NAMD files, and re-submit the simulation and connect to it again if you wish to have more time.

7.3   Timeline Analysis

We will now analyze the MDFF simulation by calculating the cross-correlation coefficient for each frame of the trajectory. As of VMD version 1.9.2, the Timeline analysis plugin found in the Analysis section of the VMD extensions menu
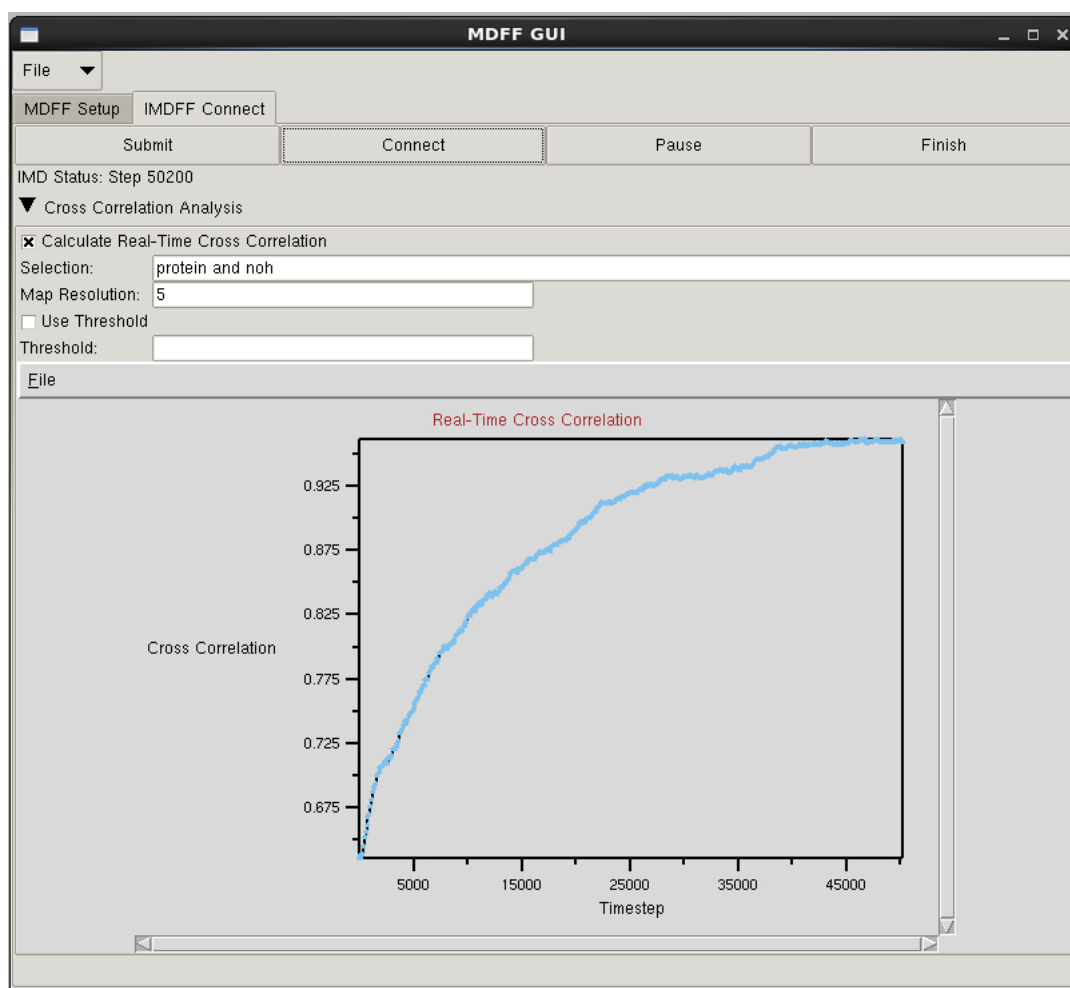
Figure 20: Analyze your interactive MDFF simulation in real-time by computing the cross correlation of the structure to the density map as the simulation runs

contains a method for quickly computing the cross correlation of an MDFF trajectory. The default behavior of this function computes the cross correlation for all the contiguous sections of secondary structure found in your model. The result is local cross correlations of the structure over the course of the entire fitting trajectory. This type of analysis provides an unprecendented level of fine-grained detail that you can use to easily examine your fitting and find regions of interest (e.g. regions of poor fit).

1 First, make sure you have your structure loaded into VMD:

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

Then, add the dcd file from your simulation:

```
mol addfile mdff-step1.dcd
```

or you may use the provided dcd's:

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

2 Open Timeline from the VMD menu in Extensions -> Analysis -> Timeline. You should see a window like the one in Fig. 21. Make sure the number on the left-hand side of the window next to the Molecule: label is the same as the ID of the molecule that contains the DCD you wish to analyze.

3 Open the Calculate menu, near the top of the Timeline window, which contains a wide range of analysis methods which you can use on your trajectory. For now, we are just interested in the 'Calc. cross-corr. ...' option. Select this method, which will open a window for parameters.

4 You should now see a window containing options for setting the parameters of the cross correlation calculation, similar to the one in Fig. 22. First you must select the density map used earlier for the MDFF fitting. This should be the same density map you provided in the MDFF Setup GUI (4ake-target_autopsf.situs), and not the resulting potential file. The next two options, map resolution and spacing, control how the simulated density is computed. The simulated density is made from your structure, which is then compared to the experimental density you fit the structure into. The resolution should be the same as the resolution of the experimental density. For the tutorial, you can leave both default values.

5 The next option, Use a map threshold, allows you to set a threshold value at which any values in the simulated density that are lesser than this value will not be included in the correlation calculation. Setting a threshold value causes the correlation to disregard lower density regions which tend to be further out from the structure. To get an idea of how different threshold values affect the resulting density map, you can use the following command in the TkConsole:

```
mdffi cc
```

Typing this in the TKConsole will print usage information. The `mdffi cc` function can be used to calculate the cross correlation, but here we are mostly interested in examing the output maps. You will need to provide an atom selection for the region of the structure you wish to use to create the map. Assuming your structure is the top molecule, you can type:
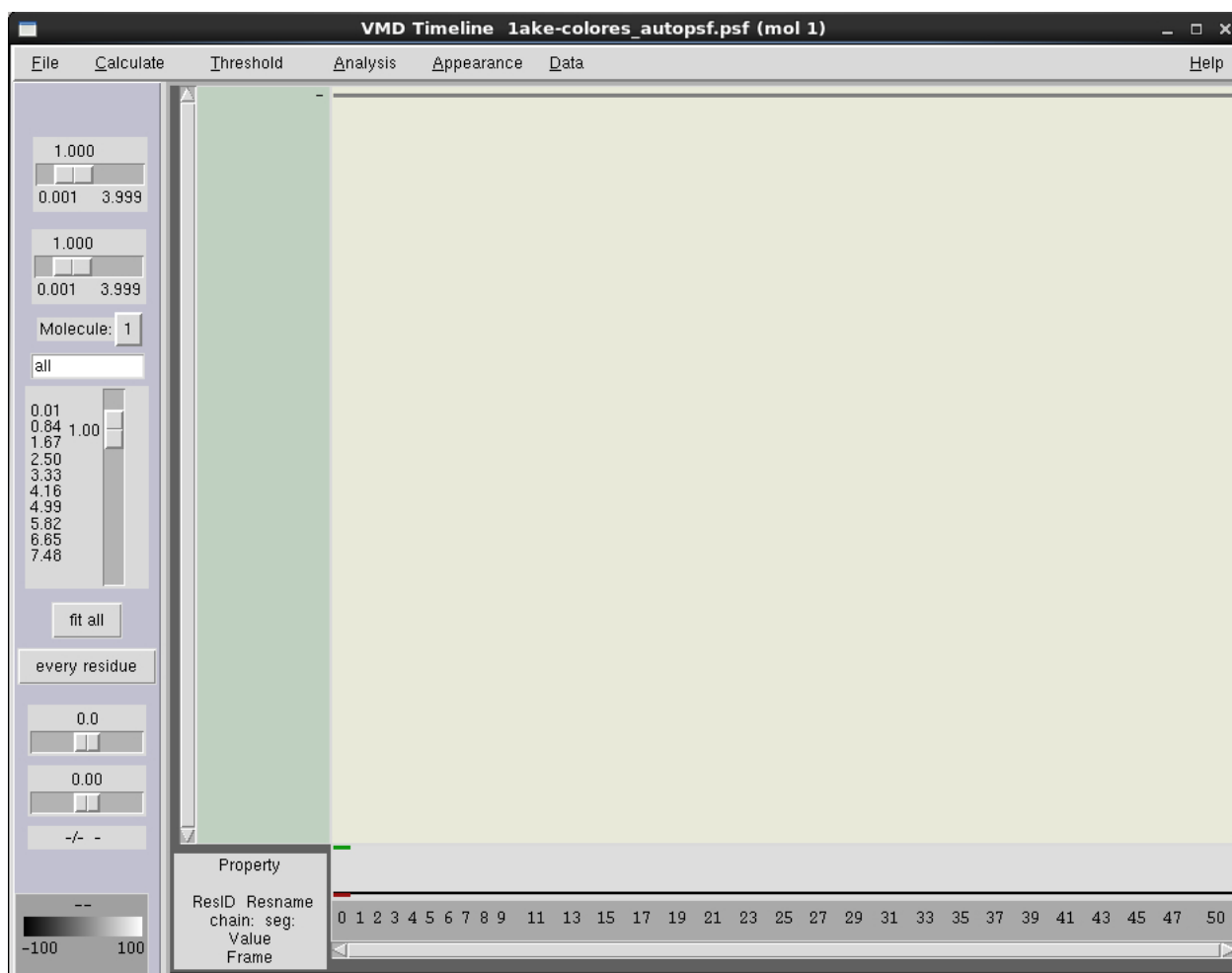
Figure 21: Timeline window prior to analysis

```
set sel [atomselect top "protein and noh"]
```

then run the mdffi cc command:

```
mdffi cc $sel -res 5 -i 4ake-target_autopsf.situs -thresholddensity
.1 -savesynthmap simulated.dx
```

This will output a simulated density map, simulated.dx, which you can
then load into VMD. You can adjust the isovalue from the VMD Repre-
sentations menu and visualize the extent of the map. You can try this for
a range of threshold values to see how the generated maps are affected.
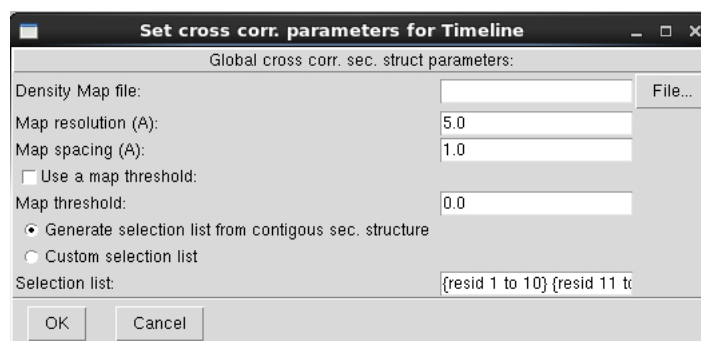For this tutorial, we can use a threshold value of 0.1.

Figure 22: Cross correlation parameters

6 There are two ways we can perform the correlation analysis. The default behavior is to segment the structure into pieces based on contiguous regions of secondary structure. Each one of these segments will have its own correlation calculated. If, however, you want to segment the structure differently, you can use the Custom selection list option and provide a set of atomselection keywords. For this tutorial we will use the default secondary structure method.

7 Select the OK button at the bottom of the parameters window. This will begin the correlation calculation and a progress window will appear. Once the calculation is complete, your Timeline window should look similar to Fig. 23.

8 The y axis of the plot contains the residues from your structure, while the x axis is the trajectory frames. Every residue in every frame has an associated colored block, which represents the local correlation of that residue to the experimental density. The color scale can be found in the bottom left corner of the Timeline window. The default view shows black squares as the worst correlation, while white is the highest. Clicking anywhere on the plot will select that residue for that particular frame and show its information in the area directly to the right of the color scale.

9 A black to white color scale can be difficult to interpret, so we will try changing to a red, green, and blue scale. To do this, click on the Appearance menu at the top of the Timeline window and select Color scale -> Rainbow (RGB), making your plot look similar to Fig. 24. Now the worst correlation will appear in red, medium correlation in green, and the best in blue. Hopefully, by the last frame, most of your structure should be blue and green.
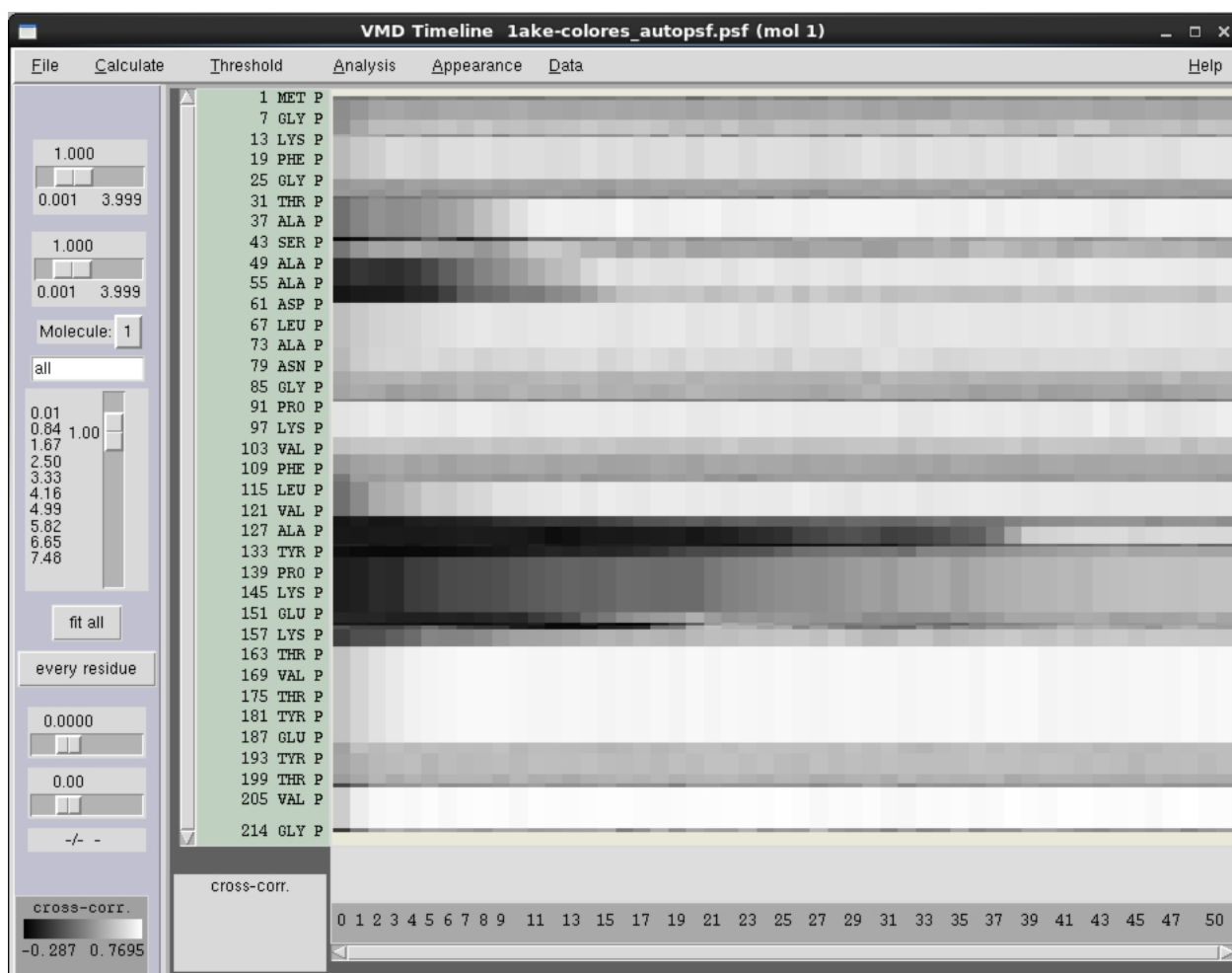
Figure 23: Timeline window after correlation analysis. Residues on the y axis, trajectory frames on the x axis

10 Next we will copy the correlation data to our structure, for another way of visualizing the correlation. From the Analysis menu at the top of the Timeline window, select 'Copy data to User field...'. This function will add the correlation data to the 'User' representation field. From the Representations menu of VMD, located in the Display menu of the VMD Main window, you should select the molecule that you analyzed in Timeline. Set the Drawing Method to NewCartoon, then set the Coloring Method to Trajectory -> User -> User. This will color the structure by the local correlations calculated in Timeline.
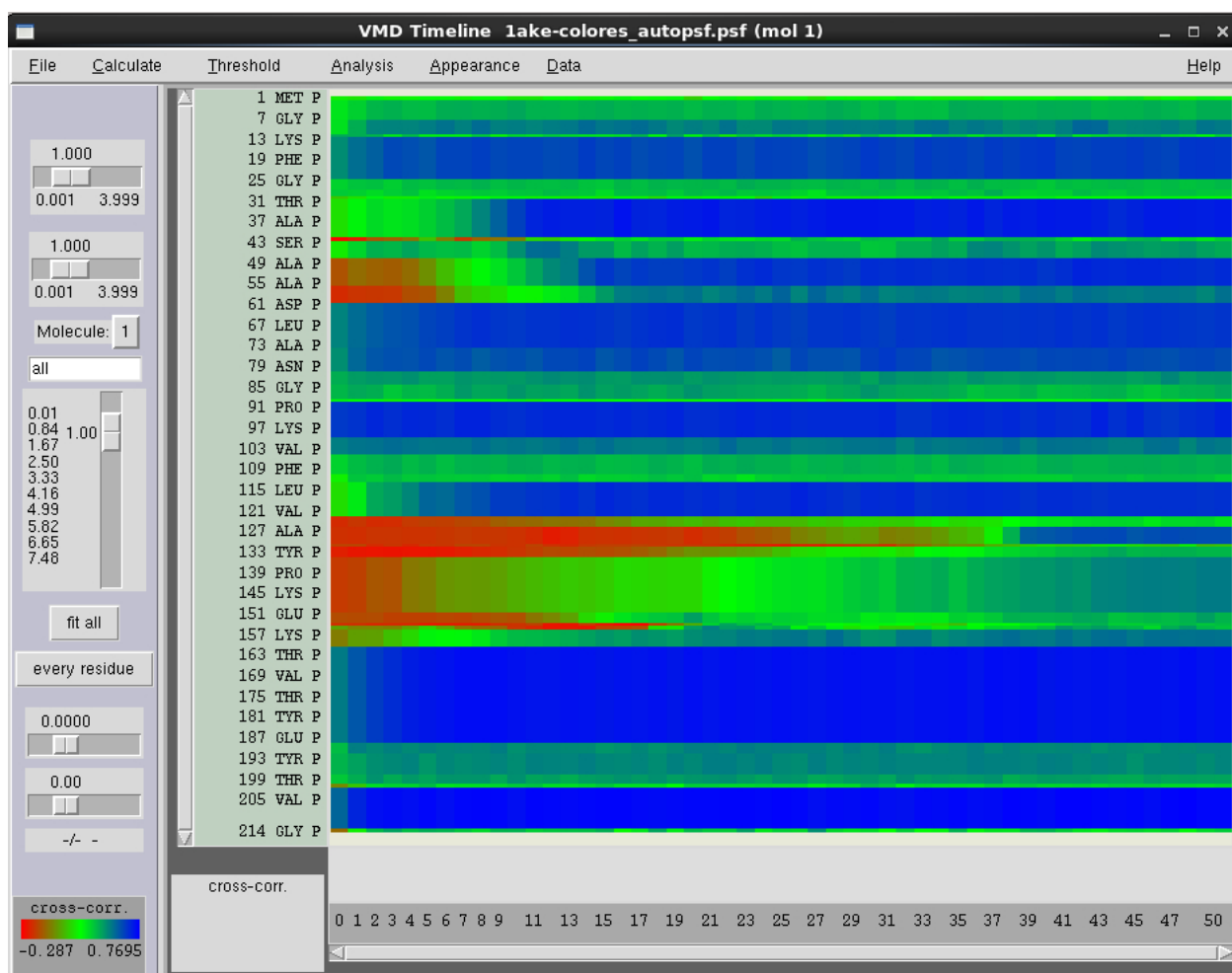
Figure 24: Timeline correlation analysis colored with red, green and blue. Red represents the worst correlation, green a medium correlation, and blue represents the best correlation

11 Because we changed our Timeline color scale to RGB, we should do the same for our representation. From the Colors window in the Graphics menu of the VMD main window, you should select the Color Scale tab at the bottom, as seen in Fig. 25. In the Method box, select RGB. Now the colors in our Timeline plot should match the colors on the structure.

12 Back in the Graphical Representations window, you should select the density map 4ake-target_autopsf.situs, and change the Draw option to
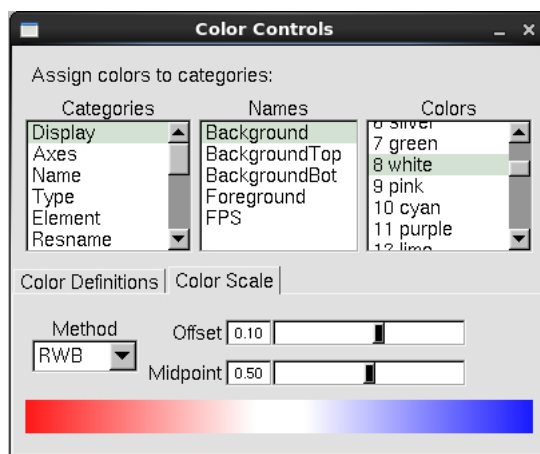
Figure 25: VMD color scale window

Solid Surface and the Material to Transparent. This will make it easier to visually inspect the fitting of your structure inside the density.

13 Whenever we select a residue in the Timeline window, that residue will be highlighted with a licorice representation in the VMD display. The trajectory will also jump to the selected frame, and the structure will be colored according to the correlations for that frame. Finding interesting events in the Timeline window and viewing the structure at that point at the same time can be a powerful method for analyzing your fitting simulation and understanding what occurred. This analysis method is especially useful for tracking down regions of poor fit and inspecting how those regions are fitting, or failing to fit, into the density map. You could set up subsequent MDFF simulations and concentrate on improving those regions, perhaps even using interactive MDFF to manually steer those regions into the density.

Acknowledgements