VMD Workshop

VISUALIZATION AND ANALYSIS OF MD TRAJECTORIES

Problems to solve

Analysis of 3.6-ns trajectory of an O_2 molecule diffusing within Mb (together):

- Make a picture of myoglobin (Mb) crystallized under Xe pressure (PDB 2W6W) using different drawing and coloring methods (pic1)
- Make a picture of all positions of the O_2 molecule diffusing within Mb for 3.6 ns (pic2a)
- Make a picture of O_2 density within Mb averaged over the 3.6-ns trajectory (pic3a)
- Make a movie of the 3.6-ns diffusion of the O_2 molecule within Mb (movie1)

Analysis of 48-ns trajectory of an O_2 molecule diffusing within Mb (self-practice):

- Find time of the O_2 escape from Mb and residues at the escape portal
- Make a picture of all positions of the O_2 molecule diffusing within Mb for 48 ns and show residues at the escape portal (pic2b)
- Make a picture of O_2 density within Mb averaged over the 48-ns trajectory and compare the regions of high O_2 population with the experimental Xe cavities (see pic1 as a reference) (pic3b)
- Plot the opening of the escape portal vs time and compare with its opening at time of the O_2 escape (estimate the opening of the portal as the area of triangle between three C_α atoms of the residues lining the portal) (plot1).

1. Starting VMD

General molecular visualization

- reads data files using an extensible plugin system,
- supports Babel for conversion of other formats.

Visualization of dynamic molecular data

- load atomic coordinate trajectories from AMBER, Charmm, DLPOLY, Gromacs, MMTK, NAMD, X-PLOR, and others.

Visualization of volumetric data

- load, generate, and display, volumetric maps

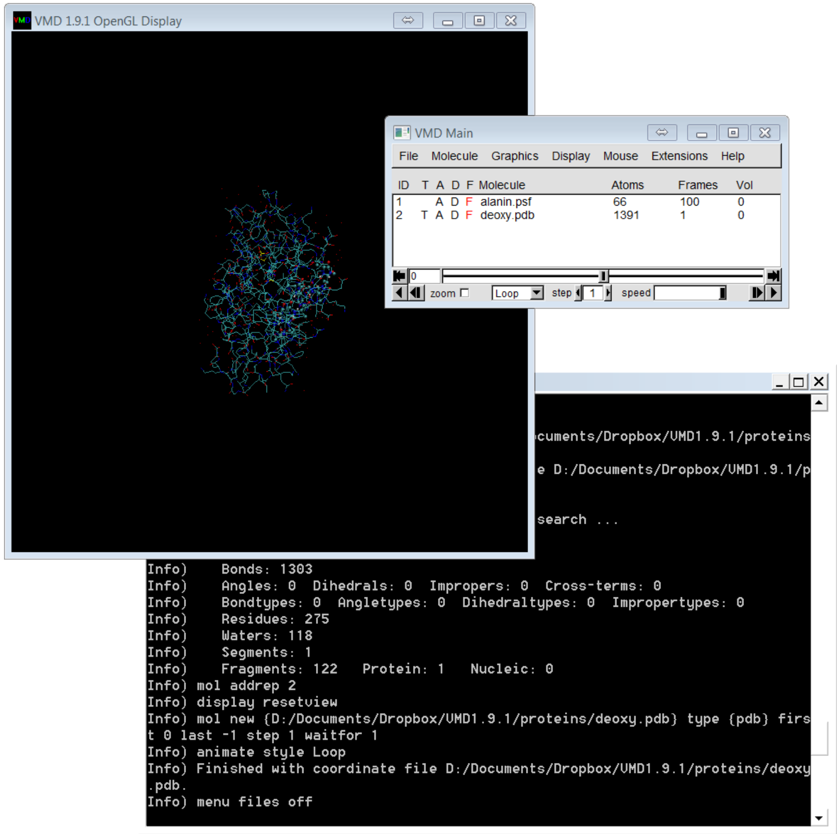
Interactive molecular dynamics simulations

- interactively apply and visualize forces in an MD simulation as it runs

Molecular analysis commands

Tcl and Python scripting languages

1.1. Molecule manipulation



```
ID  T  A  D  F  Molecule      Atoms   Frames   Vol
1      A  D  F  alanin.psf    66      100      0
2   T  A  D  F  deoxy.pdb     1391    1        0
```

```
cuments/Dropbox/VMD1.9.1/proteins

e D:/Documents/Dropbox/VMD1.9.1/p


search ...

Info)    Bonds: 1303
Info)    Angles: 0  Dihedrals: 0  Impropers: 0  Cross-terms: 0
Info)    Bondtypes: 0  Angletypes: 0  Dihedraltypes: 0  Impropertypes: 0
Info)    Residues: 275
Info)    Waters: 118
Info)    Segments: 1
Info)    Fragments: 122   Protein: 1   Nucleic: 0
Info) mol addrep 2
Info) display resetview
Info) mol new {D:/Documents/Dropbox/VMD1.9.1/proteins/deoxy.pdb} type {pdb} firs
t 0 last -1 step 1 waitfor 1
Info) animate style Loop
Info) Finished with coordinate file D:/Documents/Dropbox/VMD1.9.1/proteins/deoxy
.pdb.
Info) menu files off
```

VMD OpenGL Display

- display and manipulate molecules

VMD main menu

- manipulate molecules and trajectories
- run interfaces and extensions

VMD console

- show info and run text commands

1.1. Molecule manipulation

File → New Molecule... → load a crystal structure of Mb under Xe pressure from web (Filename: 2W6W; Determine file type: Web PDB Download)

- Press R for *rotate* mode (use 🖱 🖱 🖱 🖱 and check the VMD console)
- Press T for *translate* mode (use 🖱 🖱 🖱 🖱 and check the VMD console)
- Press S for *scale* mode (use 🖱 🖱 🖱 🖱 and check the VMD console)
- Press C to change *center* of rotation/scale
- Press 0 to get *info* about atom (check the VMD console)
- Press 1 to *label* atom
- Try 2 - 4 to *measure* distance, angle and dihedral angle
- Try 5 - 8 to *move* atom, residue, fragment and molecule

Go to Mouse →

more details at http://www.ks.uiuc.edu/Research/vmd/current/ug/node33.html

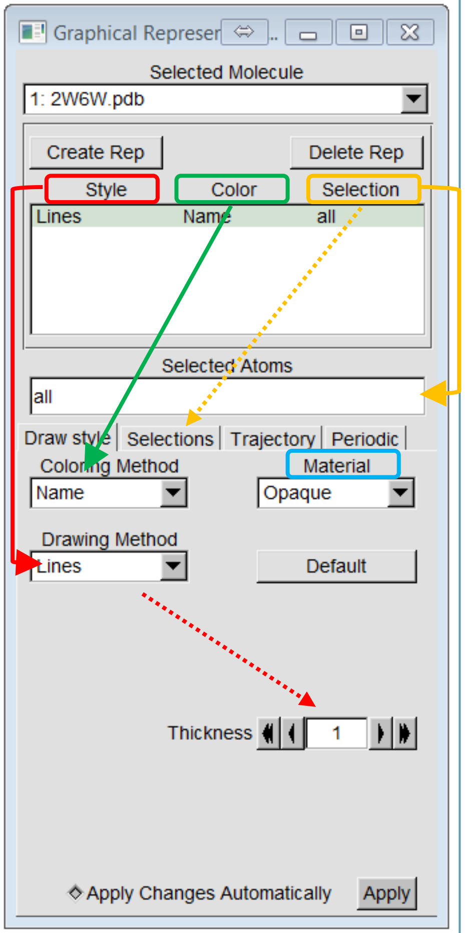
1.2. Molecule display

Graphics → Representations...

- create representations using atom selection, drawing method and coloring method

1. list of molecules

2. list of representations



Selected Molecule: 1: 2W6W.pdb

Style	Color	Selection
Lines	Name	all

Selected Atoms: all

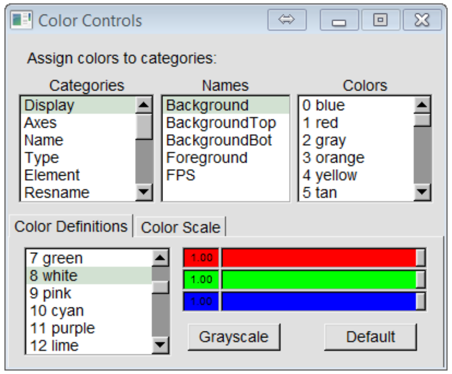
Coloring Method: Name

Material: Opaque

Drawing Method: Lines

Thickness: 1

◇ Apply Changes Automatically



Assign colors to categories:

Categories	Names	Colors
Display	Background	0 blue
Axes	BackgroundTop	1 red
Name	BackgroundBot	2 gray
Type	Foreground	3 orange
Element	FPS	4 yellow
Resname		5 tan

7 green
8 white
9 pink
10 cyan
11 purple
12 lime

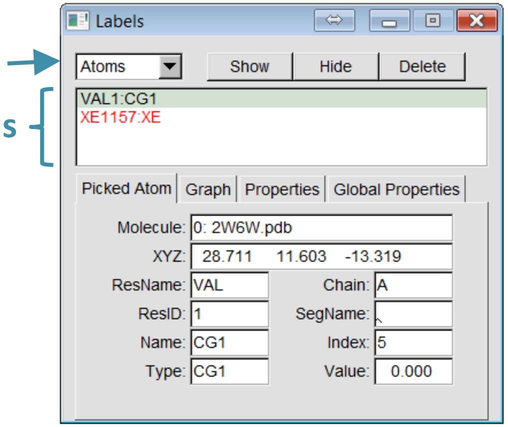
1.00
1.00
1.00

Graphics → Colors...

- assign colors to all categories

1. label types

2. list of labels



Atoms

VAL1:CG1
XE1157:XE

Molecule: 0: 2W6W.pdb

XYZ: 28.711 11.603 -13.319

ResName: VAL Chain: A

ResID: 1 SegName:

Name: CG1 Index: 5

Type: CG1 Value: 0.000

Graphics → Labels...

- manipulate labels

1.2. Molecule display

Selection examples:

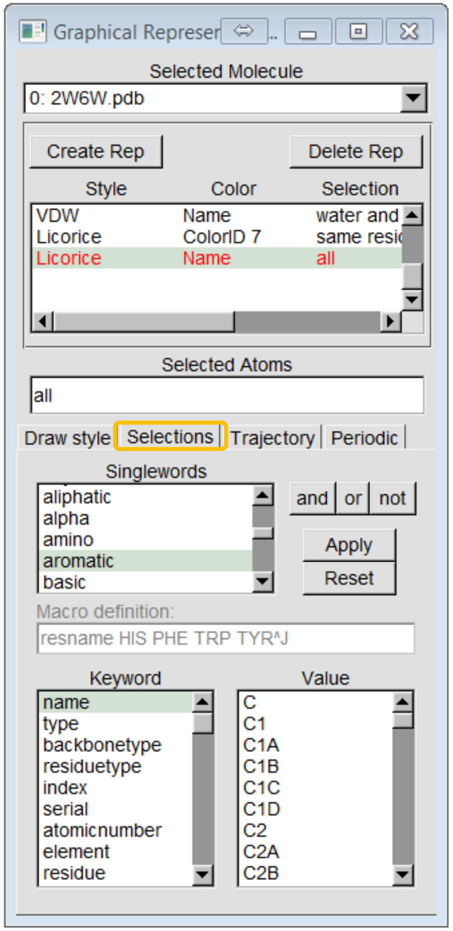
name CA
resid 35 and noh
name CA CB and resname ALA ARG
backbone and resid 1 to 6
not protein
protein (backbone or name SD)
name "C.*"
mass > 50
numbonds = 2
abs(charge) > 1
x > 30 and x < 40
sqr(x-33)+sqr(y-10)+sqr(z-7) < sqr(15)
within 10 of name FE
exwithin 3 of protein
protein within 5 of name FE
same resid as (protein within 5 of name FE)
protein sequence "K.K"

1. list of molecules

2. list of representations

singlewords

keywords and corresponding lists of values



Style	Color	Selection
VDW	Name	water and
Licorice	ColorID 7	same resi
Licorice	Name	all

Selected Atoms
all

Macro definition:
resname HIS PHE TRP TYR^J

Keyword	Value
name	C
type	C1
backbonetype	C1A
residuetype	C1B
index	C1C
serial	C1D
atomicnumber	C2
element	C2A
residue	C2B

more details at http://www.ks.uiuc.edu/Research/vmd/current/ug/node89.html

1.2. Molecule display

(1) Try Display → Reset View, Orthographic/Perspective, Depth cueing (what do they do?)

(2) Show protein backbone with coordinates of z>15 and y>4 as yellow tube (radius = 0.1)

(3) Show rest protein backbone as NewRibbons coloured by secondary structure

(4) Find and show as red Licorice all acidic residues among residues 1-20

(5) Show heme molecule as CPK colored by atom name

(6) Find atoms heavier than sulphur and show them as VDW (sphere scale = 0.5) coloured by mass

(7) Find an internal water molecule (near Fe) and show it as VDW (sphere scale = 0.5)

(8) Show residues, those atoms closer than 5 Å to the internal water, as orange licorice

(9) Label distance between the internal water and the closest Xe atom (red color, text size = 1.2, text thickness = 3)

(10)Show external water molecules as Solvent

(11)Build a protein's volumetric surface using Surf as drawing method and Glass1 as material and color it by atom name

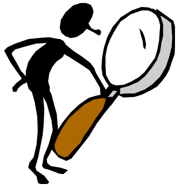
(12)Change background color to white and carbon atom color to green

1.3. Molecule scene rendering

File → Save Visualization State...

- save the visualization state as VMD file

File → Render...



- render the current scene using Snapshot (pic1.bmp)
- render the current scene using Tachyon (pic1.dat)
- render the current scene using VRML 2.0 (pic1.wrl)

low quality image

high quality image

3D interactive vector graphics

more details at http://www.ks.uiuc.edu/Research/vmd/current/ug/node111.html

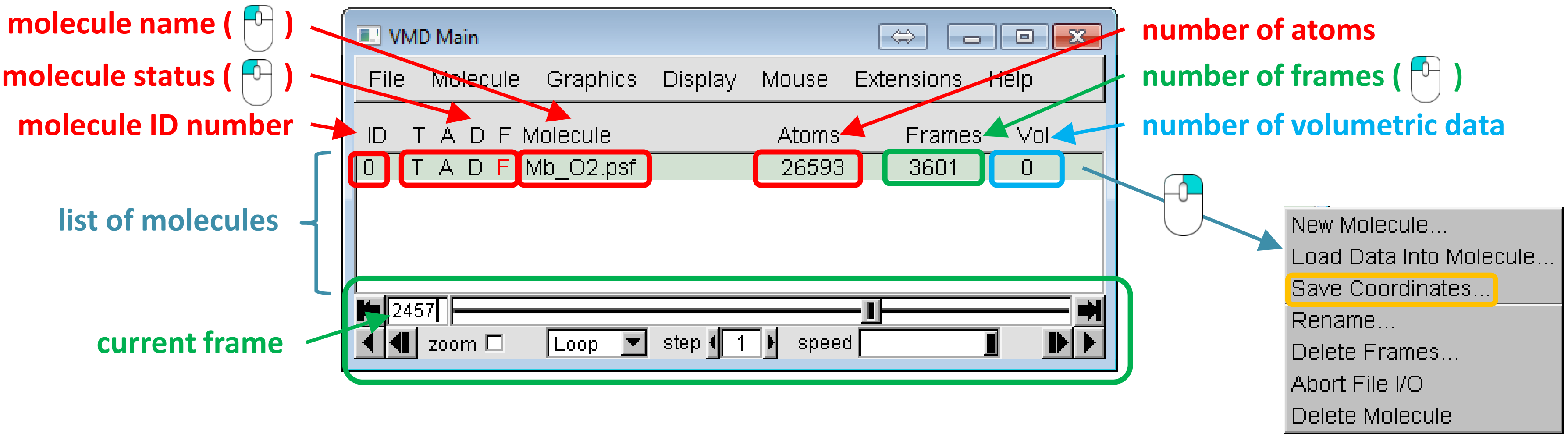
1.4. Working with MD trajectories

File → New Molecule... → Browse... → C:/cermm/VMD_workshop/Mb_O2.psf ← protein structure file

File → Load Data Into Molecule... → Browse... → C:/cermm/VMD_workshop/Mb_O2.pdb ← crystal coordinates (frame 0)

File → Load Data Into Molecule... → Browse... → C:/cermm/VMD_workshop/traj1.dcd ← MD trajectory (frames 1-3600)

- Look at the VMD console for the information about the molecule loaded



ID	T A D F	Molecule	Atoms	Frames	Vol
0	T A D F	Mb_O2.psf	26593	3601	0

molecule name ( )

molecule status ( )

molecule ID number

list of molecules

current frame

number of atoms

number of frames ( )

number of volumetric data

1.5. Analysis of MD trajectories

(1) Try Graphics → Representations... → Periodic (what can it be used for?)

(2) Using Extensions → Analysis → RMSD Trajectory Tool:

- align frames by positions of C_α atoms of protein (Trace) using crystal structure (frame 0) as a reference
- plot RMSD of C_α atoms vs frame (check Plot to make a plot with MultiPlot console)
- Note: TkConsole interactively shows data from MultiPlot

(3) Hide water, show protein as tube, heme molecule as Licorice and O_2 molecule as CPK

(4) Label the distance between the O_2 molecule and the Fe atom

(5) Plot the distance vs frame using Graphics → Labels... (at what time does O_2 diffuse from the heme cavity to the neighbouring cavity?)

1.5. Analysis of MD trajectories

(6) Create a new representation for the O_2 molecule as lines

(7) Draw multiple frames typing 0:3600 in Graphics → Representations... → Trajectory

(8) Color the representation according Timestep of the trajectory

(9) Using Extensions → Visualization → Color Scale Bar, add a heat bar for 0 to 3600 frames (autoscale off, 4 axis labels, Decimal), corresponding Timestep coloring

(10)Save a picture (pic2a)

(11)Using Extensions → Analysis → VolMapTool, create a density volumetric map of the O_2 molecule (only!) averaged over all frames of the trajectory

(12)Find a new Isosurface representation and try different Isovalues

(13)Change to Isovalue of 0.005 (white color, wireframe, without box)

(14)Save a picture (pic3a)

1.6. Making a movie in VMD

(1) Hide all representations except protein, heme and O_2

(2) Go to Extensions → Visualization → Movie Maker

- click Help to get a link to VideoMach, a movie compression soft (it is installed)
- set up working directory, name of movie (movie1), rotational angle (0), trajectory step (10)
- choose Trajectory in Movie Settings
- press Make Movie

1.7. Extensions

Biochemistry:

Extensions → Analysis →

- Contact Map
- Hydrogen Bonds
- Salt Bridges
- Timeline Plugin

- RMSD Trajectory Tool
- RMSD Visualizer Tool

- Ramachandran Plot

- Sequence Viewer
- MultiSeq

- PropKa

General:

Extensions → Analysis →

- Collective variable analysis (PLUMED)
- NAMD Energy
- NAMD Plot
- VolMap Tool

Inorganic chemistry:

Extensions → Analysis →

- IR Spectral Density Calculator
- Radial Pair Distribution Function
- Symmetry Tool

see http://www.ks.uiuc.edu/Research/vmd/plugins/

2. Scripting with Tcl/Tk in VMD

Tcl (Tool Command Language)

- powerful and highly extensible
- easy to learn and deploy
- dynamic programming language
- uses the standard I/O commands to access disk files and web and ftp sites
- suitable for a very wide range of uses
- open source and free
- cross platform (Windows, Mac OS X, Linux)

Tk (graphical user interface toolkit)

- supports many dynamic languages
- cross platform (Windows, Mac OS X, Linux)

see https://www.tcl.tk/

2.1. Starting with Tcl/Tk

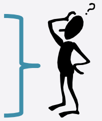
Open Extensions → Tk Console

```
#### Commands puts and set ####
#### puts value ;# creates output (in Tk Console)
puts Apple
puts Apple; puts Cake ;# to separate lines
puts -nonewline Apple; puts Cake ;# to remove new line at the end of output
puts Apple\n; puts Cake ;# to add another new line at the end of output

puts Milk and Cookies
puts "Milk and Cookies" ;# to group elements

#### set variable value ;# assigns values to variables
#### $variable  ;# refers to values of variables
#### unset variable ;# removes a variable use
set a 10
puts $a

set text Milk
puts "Glass of $text"
puts {Glass of $text} ;# to ignore $variable
```

Try ↑ ↓ in Tk Console

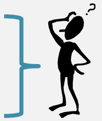
2.1. Starting with Tcl/Tk

```
#### Commands expr and relational operators ####
#### expr math_expression
expr 5/3
expr 5/3.0
```



```
expr 5%3
set a 10
expr - 3 * $a
```

```
#### eq ne || && == != < > <= >= | & ;# relational operators
expr { {apple} eq {banana} } ;# returns 1 if true, 0 if false
expr { 1 > 0}
expr {9 == 9.0}
expr {9 eq 9.0}
```



```
expr {$a>3} & {$a<30}
```

```
#### [function] ;# returns the result of function
```

```
puts "2^8 = [expr pow(2,8)]"
```

see https://www.tcl.tk/

2.1. Starting with Tcl/Tk

```
#### Commands if and for ####
#### if {expr1} then {commands} elseif {expr2} then {commands} else {commands}
if { 3.0 == 3 } {
      puts "3.0 and 3 are equal as they are numbers"
      } {
      puts "3.0 and 3 are not equal as they are strings"
      }
if { 3.0 eq 3 } {
      puts "3.0 and 3 are equal as they are numbers"
      } {
      puts "3.0 and 3 are not equal as they are strings"
      }
```

```
#### for {initialization} {test} {increment} {commands}
for {set a 0} {$a <= 10} {incr a} {
      puts "$a * 3 = [ expr $a * 3]"
      }
```

2.1. Starting with Tcl/Tk

```
#### Working with files from Tk console ####
dir
cd C:/cermm/VMD_workshop
#### open file w; open file r; close $file
#### puts $file $variable ;# creates output in a file
set file1 [open "myoutput.dat" w] ;#opens file to write
puts $file1 "All\ncats\nare\ngrey\nin\nthe\ndark"
close $file1
file exists myoutput.dat ;# returns 1 if file exits, 0 if file does exist
set file2 [open "myoutput.dat" r] ;# opens file to read
set file_data [read $file2] ;# reads data from a file
close $file2
puts $file_data
file delete myoutput.dat
```

see https://www.tcl.tk/

2.1. Starting with Tcl/Tk

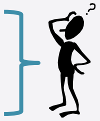
```
#### Working with lists ####
set llist {c "o" {r4 r5} duck!} ;#makes a list
llength $llist ;# returns length of the list
lindex $llist 0 ;# lists an element by index
lindex $llist 2
lindex [lindex $llist 2] 0
lappend llist {i7 i8 i9} {a} ;#add elements to the list
set llist [lreplace $llist 2 4 r d i] ;#replaces elements
set llist [linsert $llist 0 hi o n] ;#inserts elements
lset llist 0 c  ;#replaces one element
lsearch $llist c ;# returns the 1st index of element in the list
lsearch -all $llist c ;# returns all indexes of element
lsort $llist ;#sorts elements in a list
lsort -unique $llist ;#sorts a list and removes repetitions
join $llist - ;# converts list to string
```

see https://www.tcl.tk/

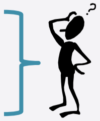
2.1. Starting with Tcl/Tk

#### Working with lists ####

```
set llist [split "1,2,3,4" ","]
set llist [split "12345" ""] ;# string to list

set llist "A B C"
puts $llist
list $llist

llength $llist
llength [list $llist]
```



```
llength [list A B C]
llength [list "A B C"]
```



2.1. Starting with Tcl/Tk

```
#### Command foreach ####
#### foreach element $list1 {commands}
set fruit_list {apples oranges grapes pears}
foreach fruit $fruit_list {
        puts $fruit
        }

#### foreach element_list1 $list1 element_list2 $list2 … {commands}
set fruit_list {apples oranges grapes pears}
set color_list {red juicy seedless Chinese}
set mass_list {2 5 1 3}
foreach fruit $fruit_list color $color_list mass $mass_list {
        puts "$mass kg of $color $fruit"
        }
```

2.2. Working with molecules using Tcl

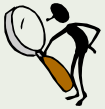
```
#### Commands mol and molinfo ####
#### mol command arguments ;# loads, modifies, or deletes a molecule in VMD

mol new Mb_O2.psf
mol addfile Mb_O2.pdb
mol addfile traj1.dcd waitfor all

#### Type mol to see a full list of its functions
```

```
#### molinfo command arguments ;# returns information about loaded molecules

molinfo num ;# number of loaded molecules
molinfo top ;# gets ID of top molecule
molinfo top get numatoms  ;# returns number of atoms
molinfo top get numframes  ;# returns number of frames
molinfo top get filename  ;# returns file names

#### Type molinfo to see a full list of its functions
```

http://www.ks.uiuc.edu/Research/vmd/current/ug/node140.html
http://www.ks.uiuc.edu/Research/vmd/current/ug/node138.html

2.2. Working with molecules using Tcl

```
#### Command atomselect ####
#### atomselect  <molid> selection ;# to access information about the atoms in a molecule
#### <molid> ↔ top ↔ (top by default)
set sel [atomselect top "protein resid 1 to 3"]
#### Type atomselect and $sel to see a full list of their functions
$sel num ;# gets number of atoms
$sel molid ;#gets selection's molecule ID
$sel text ;# gets selection's text
$sel get name ;# gets names of selection’s atoms
$sel get {resname resid} ;# gets residues names and numbers of selection’s atoms
$sel get {index name mass resname} ;# gets atom indices, names, mass and residues names
$sel get {x y z} ;# gets coordinates of selection’s atoms
$sel delete ;# deletes the selection
mol delete top ;# deletes the top molecule
```



http://www.ks.uiuc.edu/Research/vmd/current/ug/node121.html

2.3. Working with molecular trajectories via Tcl

A few examples of what we can do with tcl scripts:

(1) Measure distance between the O_2 molecule and the Fe atom vs time

(2) Measure distance between the O_2 molecule and the center of mass of protein vs time

(3) Align protein structures over trajectory (by rigid-body translations and rotations)

(4) Remove water from the trajectory

(5) Find residues, which collide with the diffusing O_2 molecule

```
#### Load the first part of the MD trajectory (traj1.dcd)

mol new Mb_O2.psf
mol addfile Mb_O2.pdb
mol addfile traj1.dcd waitfor all
```

2.3. Working with molecular trajectories via Tcl

(1) Measure distance between the O_2 molecule and the Fe atom vs time

```
set fe_sel [atomselect top "resname HEME and name FE"]
set o2_sel [atomselect top "resname O2G and name O1"]
set fe_index [$fe_sel get index]
set o2_index [$o2_sel get index]

#### measure command arguments ;# supplies algorithms for analyzing molecular structures
#### Type measure to see a full list of its functions

#### measure  bond {$index1  $index2} frame <frame>
#### measure  angle {$index1  $index2 $index3} frame <frame>
#### measure  dihed {$index1 $index2 $index3 $index4} frame <frame>
#### frame <frame> ↔ frame all ↔ (current frame by default)
#### first <frame> last <frame> step <step>

measure bond "$fe_index $o2_index" first 0 last 100 ;# distances for frames 0 - 100

set bond_list [measure bond "$fe_index $o2_index" first 0 last 100 ]
for {set i 0} {$i <= 100} {incr i} {
        puts "frame $i bond [lindex $bond_list $i]"}
```

http://www.ks.uiuc.edu/Research/vmd/current/ug/node136.html

2.3. Working with molecular trajectories via Tcl

(1) Measure distance between the O_2 molecule and the Fe atom vs time

```
#### Put data for all frames in a file

set nf [molinfo top get numframes]
set file [open "dist_o2_fe.dat" w]
puts $file "time|distance(o2-fe)"
puts $file "ns|A"

for {set i 0 } {$i < $nf } {incr i } {
        set dist [measure bond "$fe_index $o2_index" frame $i]
        set time [expr ($i/1000.0)]
        puts $file "$time|$dist"
        }

close $file
```

2.3. Working with molecular trajectories via Tcl

(2) Measure distance between the O_2 molecule and the center of mass of protein vs time

```
set o2_sel [atomselect top "resname O2G and name O1"]
$o2_sel frame 0 ;# updates selection for the frame
$o2_sel get {x y z}
$o2_sel frame 1
$o2_sel get {x y z}

set prot [atomselect top "protein"]
measure center $prot weight mass ;# returns coordinates of COM of selection at current frame

#### Measure distance between O2 and COM of protein at frame 0

$o2_sel frame 0
$prot frame 0
set o2_coord [$o2_sel get {x y z}]
set prot_center [measure center $prot weight mass]

set dist [veclength [vecsub $o2_coord $prot_center]]

#### expr ({list}) ;# to return a list without {}
set dist [veclength [vecsub [expr ($o2_coord)] $prot_center]]
```

http://www.ks.uiuc.edu/Research/vmd/current/ug/node190.html

2.3. Working with molecular trajectories via Tcl

(2) Measure distance between the O_2 molecule and the center of mass of protein vs time

```
#### Put data for all frames in a file

set file [open "dist_o2_prot.dat" w]
puts $file "time|distance(o2-prot_com)"
puts $file "ns|A"

set nf [molinfo top get numframes]

for {set i 0 } {$i < $nf } {incr i } {
        $o2_sel frame $i
        $prot frame $i
        set o2_coord [$o2_sel get {x y z}]
        set prot_center [measure center $prot]
        set dist [veclength [vecsub [expr ($o2_coord)] $prot_center]]
        set time [expr ($i/1000.0)]
        puts $file "$time|$dist"
        }

close $file
```

2.3. Working with molecular trajectories via Tcl

(3) Align protein structures over trajectory (by rigid-body translations and rotations)

```
set ca_sel [atomselect top "protein and name CA"] ;# sets up a protein selection
set ca_ref [atomselect top "protein and name CA" frame 0] ;# sets up a reference selection
set all_sel [atomselect top all] ;# sets up a selection of all atoms

set nf [molinfo top get numframes]

for {set i 0} {$i < $nf} {incr i} {
        $ca_sel frame $i ;# updates a selection
        $all_sel frame $i

        set trans_mat [measure fit $ca_sel $ca_ref] ;# measures a 4x4 transformation matrix

        $all_sel move $trans_mat ;# applies the transformation matrix to the coordinates of each
              atom in the selection

        }
```

2.3. Working with molecular trajectories via Tcl

(4) Remove water from the trajectory

```
mkdir nowater
set nowater_sel [atomselect top "protein or resname HEME O2G"] ;# sets up a new selection
$nowater_sel writepsf nowater/nowater.psf ;# creates a new psf file
set nf [molinfo top get numframes]
for {set i 0} {$i < $nf} {incr i} {
        $nowater_sel frame $i
        $nowater_sel writepdb nowater/$i.pdb ;# creates pdb files for each frame
      }
#### If you need to free memory ####
$nowater_sel delete
unset nowater_sel
```

2.3. Working with molecular trajectories via Tcl

(4) Remove water from the trajectory

```
mol load psf nowater/nowater.psf ;# loads the new psf file

#### animate command arguments ;# controls the animation of a molecular trajectory, reads
      and writes animation frames to/from a file
#### Type animate to see a full list of their functions

for {set i 1} {$i < $nf} {incr i} {
        animate read pdb nowater/$i.pdb ;# loads the new pdb files
      }

animate write dcd nowater/nowater.dcd waitfor all top ;# writes a new dcd file

mol delete top

for {set i 0 } {$i < $nf} {incr i } {
        file delete nowater/$i.pdb ;# deletes the pdb files
      }

#### Open nowater.psf and nowater.dcd and check the new trajectory
```

http://www.ks.uiuc.edu/Research/vmd/current/ug/node120.html

2.3. Working with molecular trajectories via Tcl

(5) Find residues, which collide with the diffusing O_2 molecule

```
set o2_sel [atomselect top "resname O2G"]
set o2_list [$o2_sel get index] ;# gets indexes of atoms of the $O_2$ molecule
set prot_sel [atomselect top "protein and noh"]
set prot_list [$prot_sel get index] ;# gets indexes of not-hydrogen atoms of the protein
#### Find protein residues, which are closer than 4 Å to $O_2$ at frame 0
set coll_list "" ;# set up a blank list
foreach o2_atom $o2_list { ;# runs over atom indexes of the $O_2$ molecule
        foreach prot_atom $prot_list { ;# runs over atom indexes of the protein
                set dist [measure bond "$o2_atom $prot_atom" frame $i]
                if {$dist < 4} {
                        append coll_list "␣$prot_atom" ;# adds indexes of the protein atoms to the list
                }}}
puts $coll_list
set coll_list [lsort -unique $coll_list] ;# removes repetitions from the list
#### Create a representation  with the found atoms and compare with the $O_2$ position
```

2.3. Working with molecular trajectories via Tcl

(5) Find residues, which collide with the diffusing O_2 molecule

```
#### Find protein residues, which are closer than 4 Å to O2 over the trajectory
set coll_list "" ;# set up a blank list
set nf [molinfo top get numframes]
for {set i 0} {$i < $nf} {incr i} { ;# runs over frames
    foreach o2_atom $o2_list { ;# runs over atom indexes of the O2 molecule
        foreach prot_atom $prot_list { ;# runs over atom indexes of the protein
                set dist [measure bond "$o2_atom $prot_atom" frame $i]
                if {$dist < 4} {
                        append coll_list " $prot_atom" ;# adds indexes of the protein atoms to the list
                }}}
        set coll_list [lsort -unique $coll_list] ;# removes repetitions from the list
  }
puts $coll_list
```

2.3. Working with molecular trajectories via Tcl

(5) Find residues, which collide with the diffusing O_2 molecule

```
#### Find residues corresponding to the atoms from the created index list

set coll_sel [atomselect top "index $coll_list"] ;# selects atoms from the list
$coll_sel get resid ;# finds residues numbers of the atoms
lsort -unique -real [$coll_sel get resid] ;# sorts and removes repetitions from the list of residues

#### Show the found residues as a new representation and compare with the O2
      trajectory
```

2.4. Customizing VMD

```
#### Command proc and procedures ####
#### proc name {arguments} {commands} ;# creates a new command

proc eucl_division {arg1 arg2} {
        set q [expr {$arg1/$arg2}]
        set r [expr {$arg1%$arg2}]
        return "$arg1=$arg2*$q+$r (the quotient is: $q; the remainder is: $r)"
        }

eucl_division 29 3 ;# works as a command now
```

2.4. Customizing VMD

```
#### Working with molecule representations ####
#### Remove all representations of the molecule except one
          <repid>      <molid>
mol modselect 0 top protein ;# changes the selection for a rep
mol modcolor 0 top colorid 8 ;# changes the color for a rep
mol modstyle 0 top tube 0.2 26 ;# changes the drawing style for a rep

mol addrep top ;# adds a new representation
mol modselect 1 top "resname HEME and not hydrogen"
mol modcolor 1 top colorid 1
mol modstyle 1 top licorice 0.3 30

mol addrep top
mol modselect 2 top "resname O2G"
mol modstyle 2 top lines 2
mol modcolor 2 top timestep
mol drawframes top 2 0:3600 ;# sets drawn frame range

mol delrep 2 top ;# deletes a rep

#### Type mol to see a full list of its functions
```

http://www.ks.uiuc.edu/Research/vmd/current/ug/node128.html

2.4. Customizing VMD

```
#### Drawing shapes ####
#### graphics molid command arguments = draw command arguments

mol load graphics grph ;# creates a new graphics molecule

graphics top sphere {3 3 0} radius 2 resolution 30 ;# creates a sphere of default color
graphics top color yellow ;# changes graphics color
graphics top line {6 0 0} {0 0 0} width 5 style dashed ;# creates a line of current graphics color
graphics top text {3 -3 0} "dashed line" size 2 thickness 2 ;# creates a text label

graphics top list ;# lists all graphics IDs
graphics top info 0 ;# returns info about graphics 0
graphics top delete all ;# deletes all graphics

set cyl_id [graphics top cylinder {0 0 0} {6 0 0} radius 2 resolution 30 filled 1] ;# a cylinder
graphics top delete $cyl_id
```

http://www.ks.uiuc.edu/Research/vmd/current/ug/node128.html

2.4. Customizing VMD

```
#### Changing VMD defaults ####
#### Open a file vmd.rc in the VMD directory
#### Changes turning-on of menus
menu main on ;# should be always on
menu graphics on ;# shows representations dialog
after idle { menu tkcon on }
#### Changes display defaults
display resize 600 600
axes location off
display projection orthographic
color Display Background white
#### Changes defaults for molecule representations
mol default style VDW ;# sets default style for representations (VDW not Lines)
mol default selection protein
#### Sets up user keys
user add key o {display projection orthographic}
user add key p {display projection perspective}
```

2.4. Customizing VMD

#### Working with scripts ####

```
#### Save the following strings as a tcl file (load.tcl) in the VMD directory

cd C:/cermm/VMD_workshop
mol new Mb_O2.psf
mol addfile Mb_O2.pdb
mol addfile traj1.dcd waitfor all
set nf [molinfo top get numframes]
return "$nf frames are loaded"

#### Three ways to run a tcl script:
#### 1) copy its content into TkConsole

#### 2) run VMD with -e
vmd -e load.tcl ;# starts VMD executing a specific script at startup

#### 3) source scripts from TkConsole at any time or from vmd.rc at startup
source load.tcl
```

Problems to solve

Analysis of 3.6-ns trajectory of an O_2 molecule diffusing within Mb (together):

- Make a picture of myoglobin (Mb) crystallized under Xe pressure (PDB 2W6W) using different drawing and coloring methods (pic1)
- Make a picture of all positions of the O_2 molecule diffusing within Mb for 3.6 ns (pic2a)
- Make a picture of O_2 density within Mb averaged over the 3.6-ns trajectory (pic3a)
- Make a movie of the 3.6-ns diffusion of the O_2 molecule within Mb (movie1)

Analysis of 48-ns trajectory of an O_2 molecule diffusing within Mb (self-practice):

- Find time of the O_2 escape from Mb and residues at the escape portal
- Make a picture of all positions of the O_2 molecule diffusing within Mb for 48 ns and show residues at the escape portal (pic2b)
- Make a picture of O_2 density within Mb averaged over the 48-ns trajectory and compare the regions of high O_2 population with the experimental Xe cavities (see pic1 as a reference) (pic3b)
- Plot the opening of the escape portal vs time and compare with its opening at time of the O_2 escape (estimate the opening of the portal as the area of triangle between three C_α atoms of the residues lining the portal) (plot1).

Problems to solve

Heron's formula:

the area of a triangle whose sides have lengths *a*, *b*, and *c* is

$$A = \sqrt{s(s-a)(s-b)(s-c)}$$

where *s* is the semiperimeter of the triangle; that is,

$$s = \frac{a+b+c}{2}.$$