Instituto Politécnico Nacional

Escuela Superior de Cómputo

Bioinformatics

Practice 4 - Scripting in VMD

Group: 3CV9

Student:
Ramos Diaz Enrique

Professor:
Rosas Trigueros Jorge Luis

Development Date:
October 21st 2020

Due Date:
October 28th 2020

1 Theoretical Framework

1.1 Tcl Scripting

VMD provides embedded scripting languages (Python and Tcl) for the purpose of user extensibility. Everything that can be done in VMD interactively can also be done with Tcl commands and scripts, the extensive list of Tcl text commands can help to investigate molecule properties and perform analysis [1].

Tlc is a rich language that contains many features and commands, in addition to the typical conditional and looping expressions. Tk is an extension to Tcl that permits the writing of graphical user interfaces with windows and buttons, etc [1].

Anything that can be done in the VMD graphical interface can be done with text commands. This allows scripts to be written that can automatically load molecules, create representations, analyze data, make movies, etc [1].

1.2 Examples of atomic properties

There are commands that can be used to learn about the properties (number of atoms, coordinates, total charge, etc) of an atom selection. Also there are commands to change its coordinates and other properties [1].

Many of these properties can be obtained and set using atomic selections, including segment, chain, residue, atom name, position (x, y and z), charge, mass, occupancy and radius [1].

1.3 Identifying hydrophobic residues

Many times in studies of proteins it is important to identify the location of the hydrophobic residues, as they often have a functional implication [1].

For example, in the ubiquitin the hydrophobic residues are almost exclusively contained in the inner core of the protein. This is a typical feature for small water-soluble proteins. As the protein folds, the hydrophilic residues will have a tendency to stay at the water interface, while the hydrophobic residues are pushed together and play a structural role. This help the protein achieve proper folding and increases its stability [1].

1.4 Alignment to principal axes

The principal axes of molecules are well-defined directions that often correspond to important symmetries. It is often desirable to orient a molecule so that its three principal axes are aligned with the x, y, and z directions [2].

2 Material and Equipment

- VMD - Visual Molecular Dynamics Software.
- "Scripting in VMD" tutorial from the VMD's official web page [1].
- Tutorial's files [3] provided by [1]:
  - **ubiquitin.pdb**: Ubiquitin model - 1UBQ PDB file [4].
  - **beta.tcl**
- Human methionine model - 1KQ0 PDB file [5].
- "Alignment to principal axes in VMD" script [2].
- Linear algebra packages by Hume Integration Software provided by [2]:
  - **orient.tar.gz**
  - **la101psx.tar.gz**

3 Practice Development

3.1 Following the VMD Scripting Tutorial using the Ubiquitin protein

In the lab session, the Scripting in VMD Tutorial [1] was followed using the 1UBQ - Ubiquitin protein [4] as the main model example, obtaining a visualization that highlights the hydrophobic residues of it, shown in Figure 1.

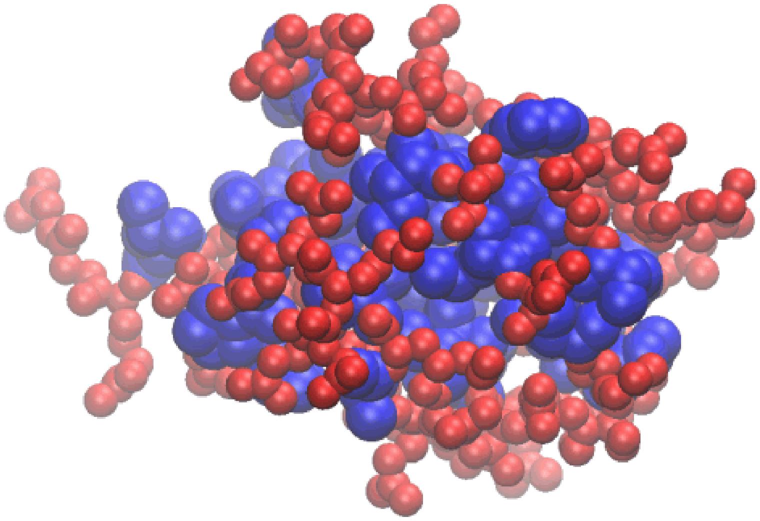


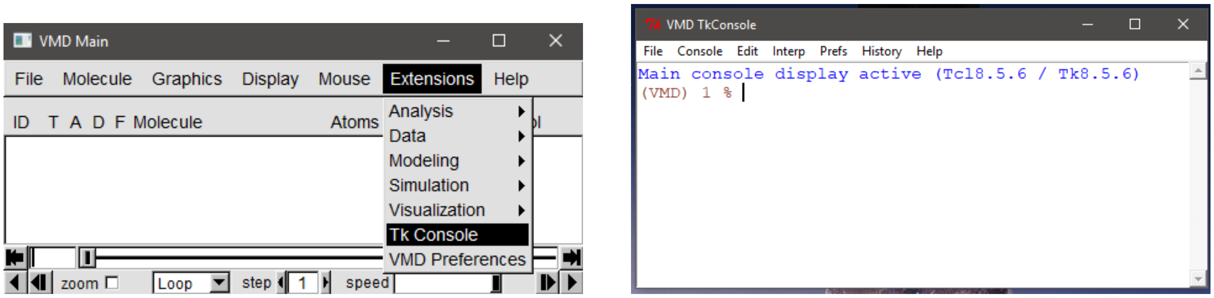
Figure 1: Ubiquitin in the VDW representation, colored according to the hydrophobicity of its residues using Tcl scripting. Image taken from [1].

Because of this, the steps of the tutorial for this specific model protein won't be reported. Instead of that, another protein model (1KQ0) will be shown and analyzed in this paper.

3.2 Following the VMD Scripting Tutorial using a different protein: 1KQ0

3.2.1 Loading molecules with text commands

Start VMD. In the VMD Main window, choose *Extensions* > *Tk Console* to open the VMD TkConsole window (shown in Figure 2).



```
Main console display active (Tcl8.5.6 / Tk8.5.6)
(VMD) 1 %
```

Figure 2: VMD TkConsole window.

Firs of all, in the TKConsole navigate using linux or windows prompt commands to the $FOLDER_PROJECT, a directory where all the files listed in the Material and Equipment are saved. Then type the command:

```
    mol new 1kq0.pdb
```

and hit enter. As can be seen in Figure 3, this command performs the same function as loading a molecule through the File Browser window.

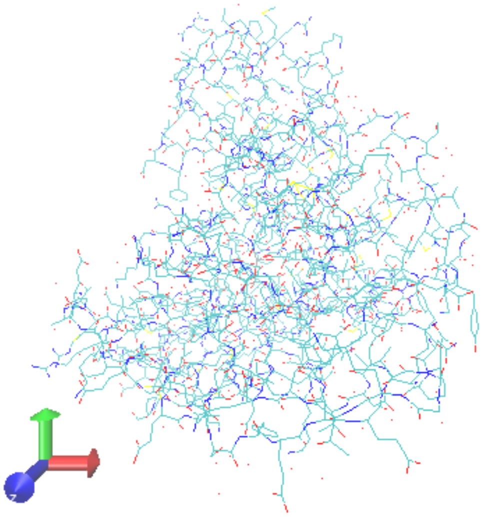
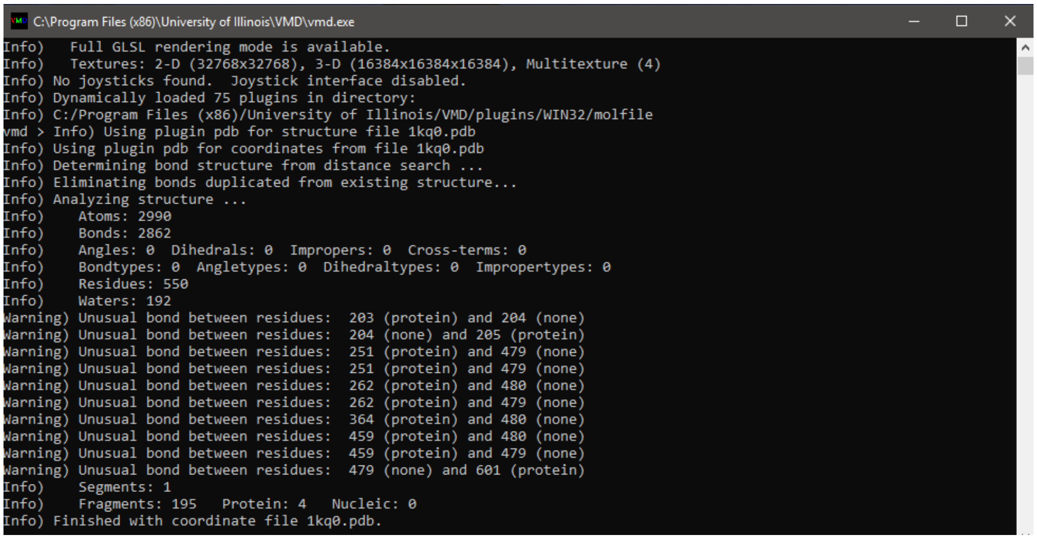


Figure 3: 1KQ0 Human methionine loaded using the VMD TkConsole.

When VMD is opened, by default a vmd console window appears, shown in Figure 4. The vmd console window tells what's going on within the VMD session that are working on. It should tell that a molecule has been loaded, as well as some of its basic properties like number of atoms, bonds, residues and etc. The Tcl commands that are entered in the VMD TkConsole window can also be entered in the vmd console window [1].



```
Info)   Full GLSL rendering mode is available.
Info)   Textures: 2-D (32768x32768), 3-D (16384x16384x16384), Multitexture (4)
Info) No joysticks found.  Joystick interface disabled.
Info) Dynamically loaded 75 plugins in directory:
Info) C:/Program Files (x86)/University of Illinois/VMD/plugins/WIN32/molfile
vmd > Info) Using plugin pdb for structure file 1kq0.pdb
Info) Using plugin pdb for coordinates from file 1kq0.pdb
Info) Determining bond structure from distance search ...
Info) Eliminating bonds duplicated from existing structure...
Info) Analyzing structure ...
Info)    Atoms: 2990
Info)    Bonds: 2862
Info)    Angles: 0  Dihedrals: 0  Impropers: 0  Cross-terms: 0
Info)    Bondtypes: 0  Angletypes: 0  Dihedraltypes: 0  Impropertypes: 0
Info)    Residues: 550
Info)    Waters: 192
Warning) Unusual bond between residues:  203 (protein) and 204 (none)
Warning) Unusual bond between residues:  204 (none) and 205 (protein)
Warning) Unusual bond between residues:  251 (protein) and 479 (none)
Warning) Unusual bond between residues:  251 (protein) and 479 (none)
Warning) Unusual bond between residues:  262 (protein) and 480 (none)
Warning) Unusual bond between residues:  262 (protein) and 479 (none)
Warning) Unusual bond between residues:  364 (protein) and 480 (none)
Warning) Unusual bond between residues:  459 (protein) and 480 (none)
Warning) Unusual bond between residues:  459 (protein) and 479 (none)
Warning) Unusual bond between residues:  479 (none) and 601 (protein)
Info)    Segments: 1
Info)    Fragments: 195   Protein: 4   Nucleic: 0
Info) Finished with coordinate file 1kq0.pdb.
```

Figure 4: VMD Console after loading the 1KQ0 model using the VMD TkConsole.

3.2.2 The atomselect command

VMD's *atomselect* command allows to select a specific part of a molecule [1].

```
atomselect molid selection
```

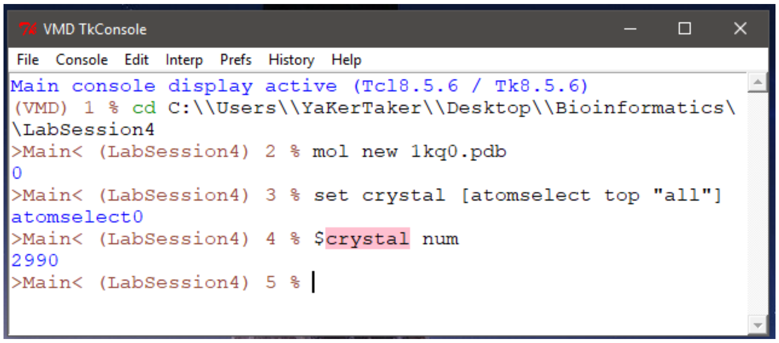
The first argument to *atomselect* is the molecule ID, the second argument is a textual atom selection like what it has been using to describe graphical representations in the Graphical Representations window [1].

Type:

```
set crystal [atomselect top "all"]
```

in the TkConsole window. This creates a selection, *$crystal*, that contains all the atoms in the molecule and assigns it to the variable crystal. Instead of a molecule ID (which is a number), the shortcut "top" have been used to refer to the top molecule.

The result of atomselect is a function. Thus, *$crystal* is now a function that performs actions on the contents of the "all" selection [1]. Figure 5 shows the result of this command in the VMD TkConsole.



```
Main console display active (Tcl8.5.6 / Tk8.5.6)
(VMD) 1 % cd C:\\Users\\YaKerTaker\\Desktop\\Bioinformatics\
\LabSession4
>Main< (LabSession4) 2 % mol new 1kq0.pdb
0
>Main< (LabSession4) 3 % set crystal [atomselect top "all"]
atomselect0
>Main< (LabSession4) 4 % $crystal num
2990
>Main< (LabSession4) 5 % |
```

Figure 5: *Atomselect* and *num* commands.

3.2.3 Obtaining and changing molecule properties with text commands

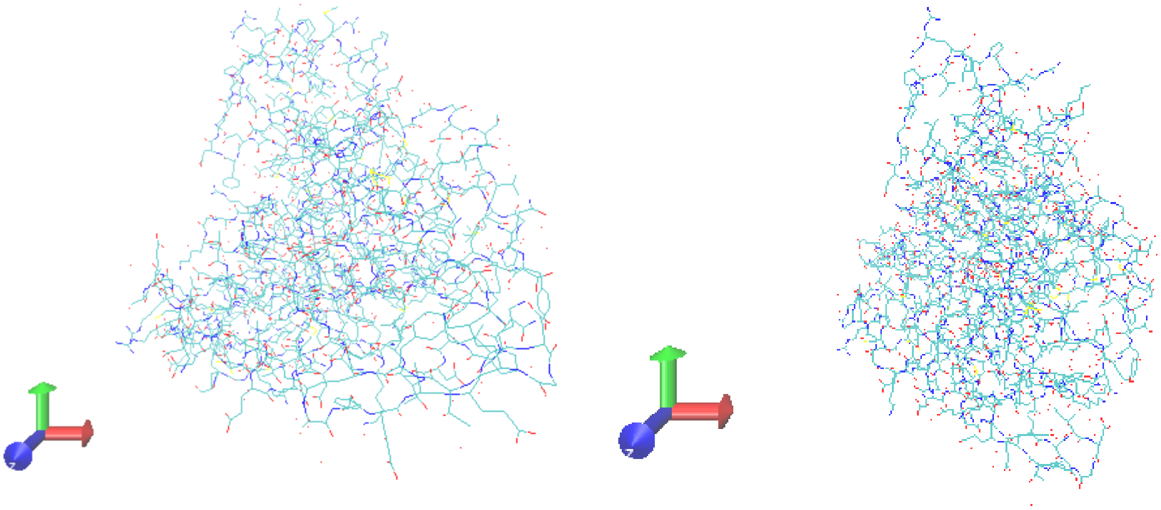
Typing:

```
    $crystal num
```

in the TkConsole window returns the number of atoms in that selection. This number matches the number of atoms for the molecule displayed in the VMD Main window (shown too in Figure 5) [1].

Additionally, there are commands to move a molecule on the screen. Using these commands to change the atom coordinates (Figure 6 shows the result of this commands):

```
    $crystal moveby {10 0 0}
    $crystal move [transaxis x 40 deg]
```



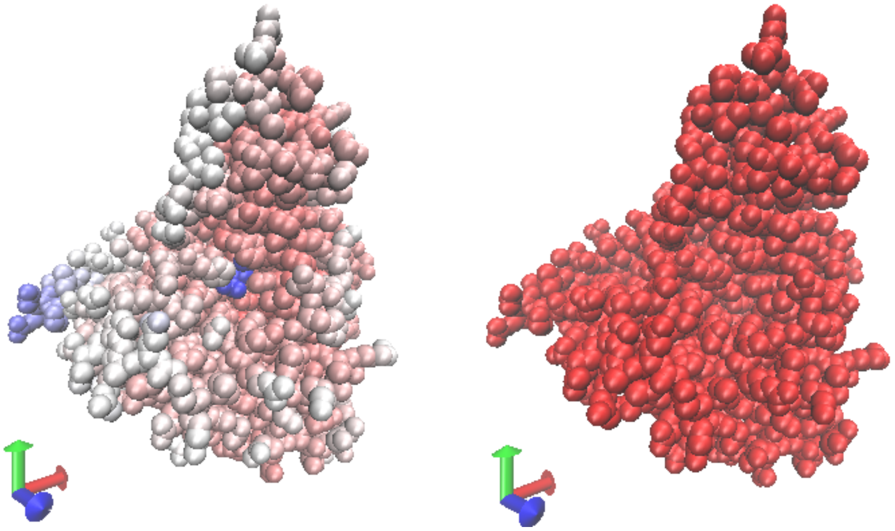
(a) $crystal moveby 10 0 0

(b) $crystal move [transaxis x 40 deg

Figure 6: Results of using commands to move the 1KQ0 on screen.

Now a visualization that highlights the hydrophobic residues of the 1KQ0 protein will be obtained.

Open the Graphical Representation window. Type in protein as the atom selection, change its Coloring Method to Beta and its Drawing Method to VDW. Figure 7a shows how the molecule should now appear as a white, red and blue assembly of spheres.



(a) 1KQ0 protein with Beta coloring method and VDW drawing method.

(b) 1KQ0 protein (*$crystal* selection) with beta set as zero.

Figure 7: Preparing the molecule to get their hydrophobic and hydrophilic residues using Tcl commands.

Back to the TkConsole window, type:

```
    $crystal set beta 0
```

This resets the "beta" field (which is displayed) to zero for all atoms. Figure 7b shows that the atoms displayed will suddenly change to a uniform color (since they all have the same beta values now) [1].

Then in the TkConsole window, type:

```
    set sel [atomselect top "hydrophobic"]
```

This creates a selection, *$sel*, that contains all the atoms in the hydrophobic residues. Let's label all hydrophobic atoms by setting their beta values to 1. In the TkConsole window, type:

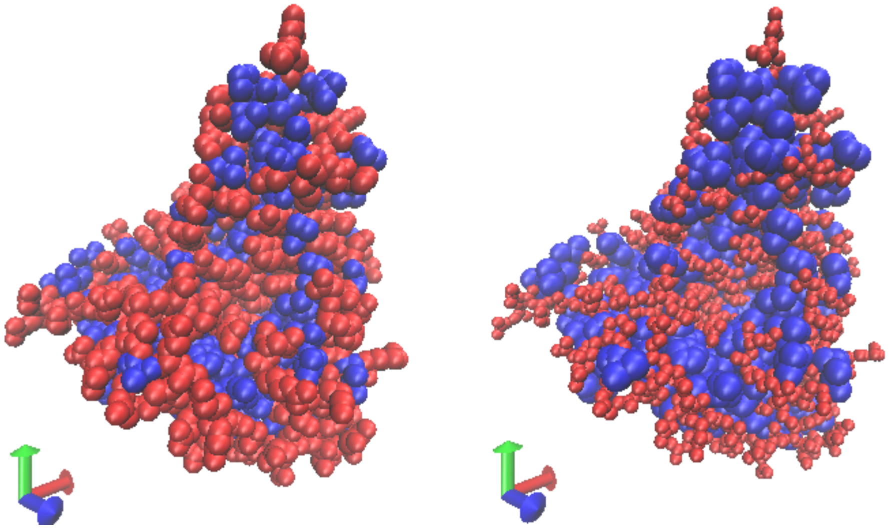
```
    $sel set beta 1
```

Figure 8a shows the result of this command. If the colors in the OpenGL Display do not get updated, go to the Graphical Representations window and click on the "Apply button" at the bottom.

Then, a physical property of the atoms will be changed to further illustrate the distribution of hydrophobic residues. To make all the atoms smaller, in the TkConsole window type:

```
      $crystal set radius 1.0
      $sel set radius 2.0
```

Figure 8b display a visual state that clearly distinguishes which parts of the 1KQ0 protein are hydrophobic and which are hydrophilic.



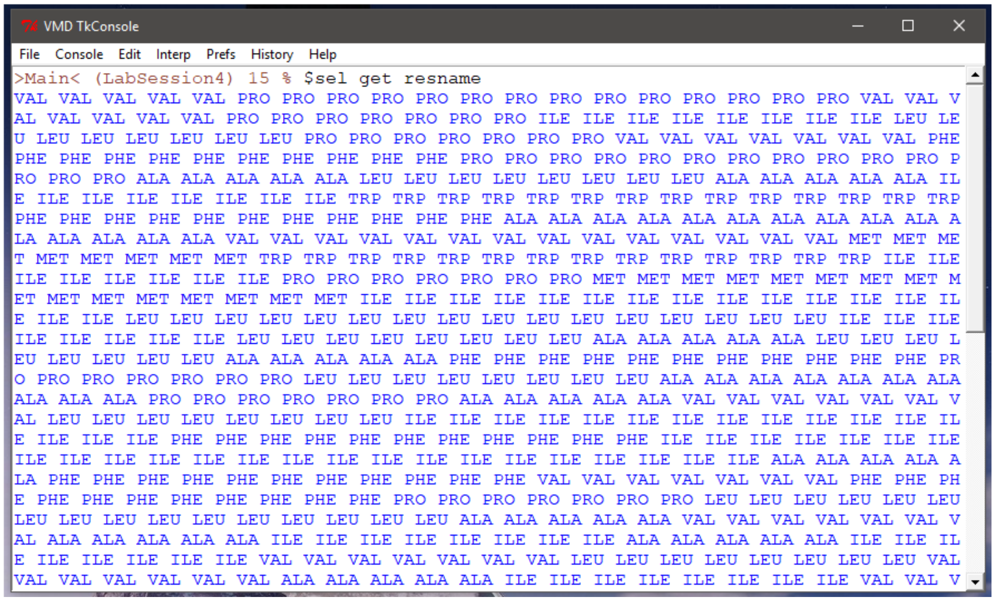
(a) 1KQ0 protein without radius set.

(b) 1KQ0 protein with radius set.

Figure 8: 1KQ0 protein colored according to the hydrophobicity of its residues using Tcl scripting.

Atom selections are useful not only for setting atomic data, but also for getting atomic information [1]. Let's say that hydrophobic residues are wanted to list. Use the *get* command with the *$sel* atom selection to obtain the names of hydrophobic residues (result of this command is shown in Figure 9):

```
      $sel get resname
```



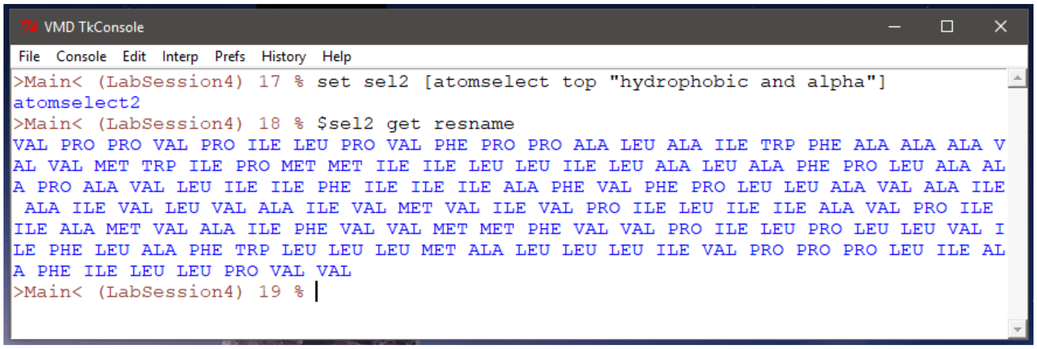
```
>Main< (LabSession4) 15 % $sel get resname
VAL VAL VAL VAL VAL PRO PRO PRO PRO PRO PRO PRO PRO PRO PRO PRO PRO PRO PRO VAL VAL V
AL VAL VAL VAL VAL PRO PRO PRO PRO PRO PRO PRO ILE ILE ILE ILE ILE ILE ILE ILE LEU LE
U LEU LEU LEU LEU LEU LEU PRO PRO PRO PRO PRO PRO PRO VAL VAL VAL VAL VAL VAL VAL PHE
PHE PHE PHE PHE PHE PHE PHE PHE PHE PHE PRO PRO PRO PRO PRO PRO PRO PRO PRO PRO PRO P
RO PRO PRO ALA ALA ALA ALA ALA LEU LEU LEU LEU LEU LEU LEU LEU ALA ALA ALA ALA ALA IL
E ILE ILE ILE ILE ILE ILE ILE TRP TRP TRP TRP TRP TRP TRP TRP TRP TRP TRP TRP TRP TRP
PHE PHE PHE PHE PHE PHE PHE PHE PHE PHE PHE ALA ALA ALA ALA ALA ALA ALA ALA ALA ALA A
LA ALA ALA ALA ALA VAL VAL VAL VAL VAL VAL VAL VAL VAL VAL VAL VAL VAL VAL MET MET ME
T MET MET MET MET MET TRP TRP TRP TRP TRP TRP TRP TRP TRP TRP TRP TRP TRP TRP ILE ILE
ILE ILE ILE ILE ILE ILE PRO PRO PRO PRO PRO PRO PRO MET MET MET MET MET MET MET MET M
ET MET MET MET MET MET MET MET ILE ILE ILE ILE ILE ILE ILE ILE ILE ILE ILE ILE ILE IL
E ILE ILE LEU LEU LEU LEU LEU LEU LEU LEU LEU LEU LEU LEU LEU LEU LEU LEU ILE ILE ILE
ILE ILE ILE ILE ILE LEU LEU LEU LEU LEU LEU LEU LEU ALA ALA ALA ALA ALA LEU LEU LEU L
EU LEU LEU LEU LEU ALA ALA ALA ALA ALA PHE PHE PHE PHE PHE PHE PHE PHE PHE PHE PHE PR
O PRO PRO PRO PRO PRO PRO LEU LEU LEU LEU LEU LEU LEU LEU ALA ALA ALA ALA ALA ALA ALA
ALA ALA ALA PRO PRO PRO PRO PRO PRO PRO ALA ALA ALA ALA ALA VAL VAL VAL VAL VAL VAL V
AL LEU LEU LEU LEU LEU LEU LEU LEU ILE ILE ILE ILE ILE ILE ILE ILE ILE ILE ILE ILE IL
E ILE ILE ILE PHE PHE PHE PHE PHE PHE PHE PHE PHE PHE PHE ILE ILE ILE ILE ILE ILE ILE
ILE ILE ILE ILE ILE ILE ILE ILE ILE ILE ILE ILE ILE ILE ILE ILE ILE ALA ALA ALA ALA A
LA PHE PHE PHE PHE PHE PHE PHE PHE PHE PHE PHE VAL VAL VAL VAL VAL VAL VAL PHE PHE PH
E PHE PHE PHE PHE PHE PHE PHE PHE PRO PRO PRO PRO PRO PRO PRO LEU LEU LEU LEU LEU LEU
LEU LEU LEU LEU LEU LEU LEU LEU LEU LEU ALA ALA ALA ALA ALA VAL VAL VAL VAL VAL VAL V
AL ALA ALA ALA ALA ALA ILE ILE ILE ILE ILE ILE ILE ILE ALA ALA ALA ALA ALA ILE ILE IL
E ILE ILE ILE ILE ILE VAL VAL VAL VAL VAL VAL VAL LEU LEU LEU LEU LEU LEU LEU LEU VAL
VAL VAL VAL VAL VAL VAL ALA ALA ALA ALA ALA ILE ILE ILE ILE ILE ILE ILE ILE VAL VAL V
```

Figure 9: *Get* command to know the hydrophobic residues in the 1KQ0 protein.

Each residue contains many atoms, resulting in multiple repeated entries, but each amino acid has the same backbone atoms. If only one of these atoms per residue is picked, each residue will be present only once in a selection [1].

Each residue has one and only one α -carbon (name CA = alpha) [1], so type the following in the TkConsole window (result of this command is shown in Figure 10):

```
    set sel2 [atomselect top "hydrophobic and alpha"]
    $sel2 get resname
```

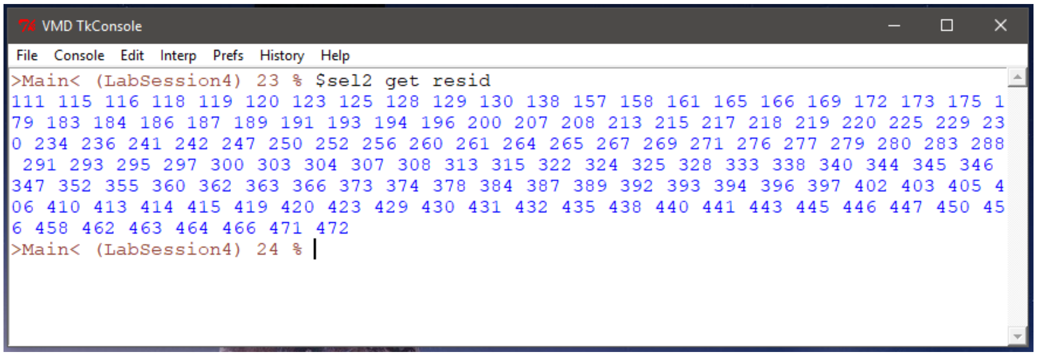


```
>Main< (LabSession4) 17 % set sel2 [atomselect top "hydrophobic and alpha"]
atomselect2
>Main< (LabSession4) 18 % $sel2 get resname
VAL PRO PRO VAL PRO ILE LEU PRO VAL PHE PRO PRO ALA LEU ALA ILE TRP PHE ALA ALA ALA V
AL VAL MET TRP ILE PRO MET MET ILE ILE LEU LEU ILE LEU ALA LEU ALA PHE PRO LEU ALA AL
A PRO ALA VAL LEU ILE ILE PHE ILE ILE ILE ALA PHE VAL PHE PRO LEU LEU ALA VAL ALA ILE
 ALA ILE VAL LEU VAL ALA ILE VAL MET VAL ILE VAL PRO ILE LEU ILE ILE ALA VAL PRO ILE
ILE ALA MET VAL ALA ILE PHE VAL VAL MET MET PHE VAL VAL PRO ILE LEU PRO LEU LEU VAL I
LE PHE LEU ALA PHE TRP LEU LEU LEU MET ALA LEU LEU LEU ILE VAL PRO PRO PRO LEU ILE AL
A PHE ILE LEU LEU PRO VAL VAL
>Main< (LabSession4) 19 % |
```

Figure 10: Avoiding multiple repeated entries while getting the hydrophobic residues in the 1KQ0 protein.

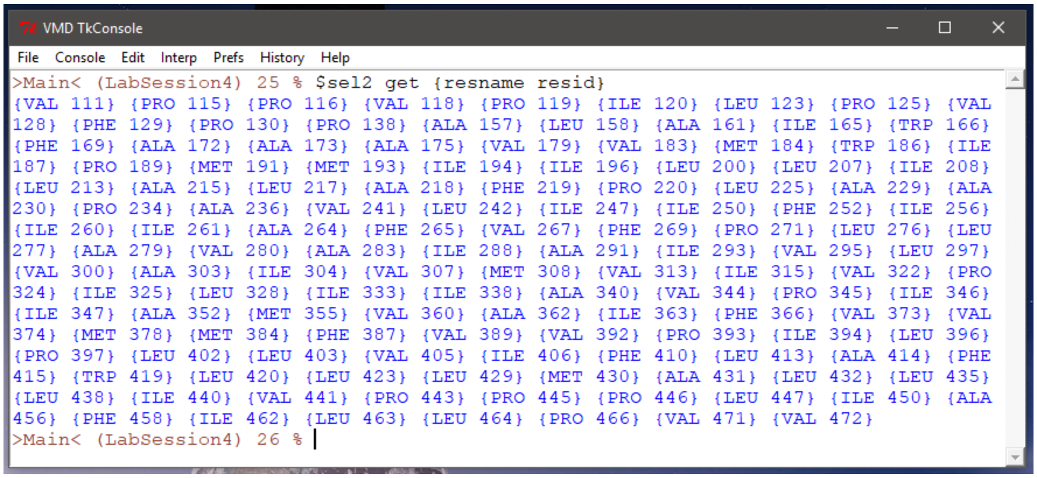
To get multiple properties simultaneously, type the following (result of this command is shown in Figure 11, 12 and 13):

```
    $sel2 get resid
    $sel2 get {resname resid}
    $sel2 get {x y z}
```



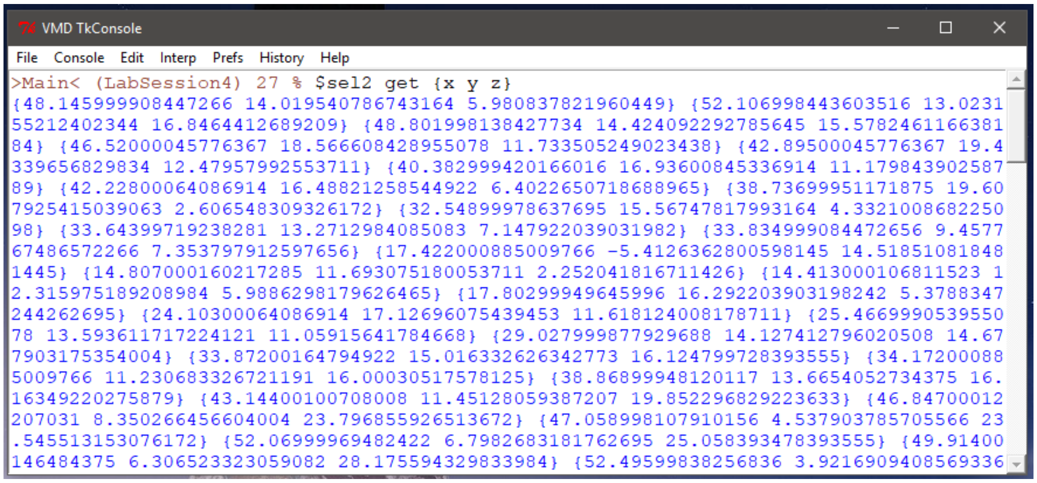
```
>Main< (LabSession4) 23 % $sel2 get resid
111 115 116 118 119 120 123 125 128 129 130 138 157 158 161 165 166 169 172 173 175 1
79 183 184 186 187 189 191 193 194 196 200 207 208 213 215 217 218 219 220 225 229 23
0 234 236 241 242 247 250 252 256 260 261 264 265 267 269 271 276 277 279 280 283 288
 291 293 295 297 300 303 304 307 308 313 315 322 324 325 328 333 338 340 344 345 346
347 352 355 360 362 363 366 373 374 378 384 387 389 392 393 394 396 397 402 403 405 4
06 410 413 414 415 419 420 423 429 430 431 432 435 438 440 441 443 445 446 447 450 45
6 458 462 463 464 466 471 472
>Main< (LabSession4) 24 %
```

Figure 11: Getting the hydrophobic residues IDs in the 1KQ0 protein.



```
>Main< (LabSession4) 25 % $sel2 get {resname resid}
{VAL 111} {PRO 115} {PRO 116} {VAL 118} {PRO 119} {ILE 120} {LEU 123} {PRO 125} {VAL
128} {PHE 129} {PRO 130} {PRO 138} {ALA 157} {LEU 158} {ALA 161} {ILE 165} {TRP 166}
{PHE 169} {ALA 172} {ALA 173} {ALA 175} {VAL 179} {VAL 183} {MET 184} {TRP 186} {ILE
187} {PRO 189} {MET 191} {MET 193} {ILE 194} {ILE 196} {LEU 200} {LEU 207} {ILE 208}
{LEU 213} {ALA 215} {LEU 217} {ALA 218} {PHE 219} {PRO 220} {LEU 225} {ALA 229} {ALA
230} {PRO 234} {ALA 236} {VAL 241} {LEU 242} {ILE 247} {ILE 250} {PHE 252} {ILE 256}
{ILE 260} {ILE 261} {ALA 264} {PHE 265} {VAL 267} {PHE 269} {PRO 271} {LEU 276} {LEU
277} {ALA 279} {VAL 280} {ALA 283} {ILE 288} {ALA 291} {ILE 293} {VAL 295} {LEU 297}
{VAL 300} {ALA 303} {ILE 304} {VAL 307} {MET 308} {VAL 313} {ILE 315} {VAL 322} {PRO
324} {ILE 325} {LEU 328} {ILE 333} {ILE 338} {ALA 340} {VAL 344} {PRO 345} {ILE 346}
{ILE 347} {ALA 352} {MET 355} {VAL 360} {ALA 362} {ILE 363} {PHE 366} {VAL 373} {VAL
374} {MET 378} {MET 384} {PHE 387} {VAL 389} {VAL 392} {PRO 393} {ILE 394} {LEU 396}
{PRO 397} {LEU 402} {LEU 403} {VAL 405} {ILE 406} {PHE 410} {LEU 413} {ALA 414} {PHE
415} {TRP 419} {LEU 420} {LEU 423} {LEU 429} {MET 430} {ALA 431} {LEU 432} {LEU 435}
{LEU 438} {ILE 440} {VAL 441} {PRO 443} {PRO 445} {PRO 446} {LEU 447} {ILE 450} {ALA
456} {PHE 458} {ILE 462} {LEU 463} {LEU 464} {PRO 466} {VAL 471} {VAL 472}
>Main< (LabSession4) 26 %
```

Figure 12: Getting the hydrophobic residues names in the 1KQ0 protein.



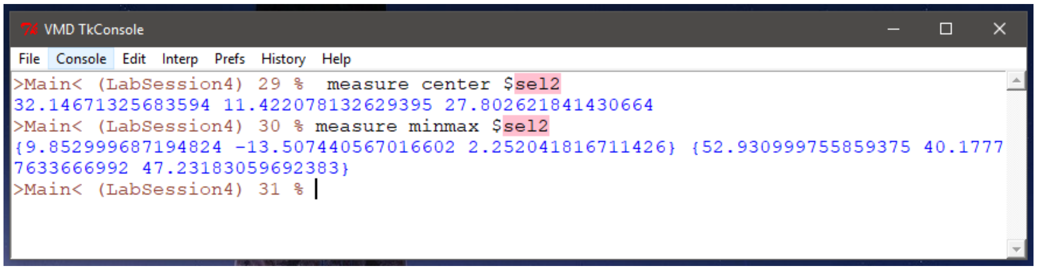
```
>Main< (LabSession4) 27 % $sel2 get {x y z}
{48.145999908447266 14.019540786743164 5.980837821960449} {52.106998443603516 13.0231
55212402344 16.846412689209} {48.801998138427734 14.424092292785645 15.57824611666381
84} {46.52000045776367 18.566608428955078 11.733505249023438} {42.89500045776367 19.4
33965682983 4 12.47957992553711} {40.382999420166016 16.93600845336914 11.17984390258 7
89} {42.2280006408 6914 16.48821258544922 6.402265071868965} {38.73699951171875 19.60
7925415039063 2.606548309326172} {32.54899978637695 15.56747817993164 4.332100868225 0
98} {33.6439971923 8281 13.2712984085083 7.147922039031982} {33.83499908447265 6 9.4577
67486572266 7.353797912597656} {17.422000885009766 -5.412636280059814 5 14.5185108184 8
1445} {14.807000160217285 11.693075180053711 2.252041816711426} {14.413000106811523 1
2.315975189208984 5.988629817962646 5} {17.802999496459 96 16.292203903198242 5.3788347
244262695} {24.10300064086914 17.12696075439453 11.618124008178711} {25.466999053955 0
78 13.593611717224121 11.05915641784668} {29.027999877929688 14.127412796020508 14.67
790317535400 4} {33.87200164794922 15.016332626342773 16.124799728393555} {34.17200088
5009766 11.230683326721191 16.00030517578125} {38.86899948120117 13.665405273437 5 16.
163492202758 79} {43.144001007080 08 11.451280593872 07 19.852296829223633} {46.84700012
207031 8.350266456604004 23.796855926513672} {47.058998107910156 4.537903785705566 23
.545513153076172} {52.06999969482422 6.798268318176269 5 25.058393478393555} {49.91400
146484375 6.306523323059082 28.175594329833984} {52.49599838256836 3.9216909408569336
```

Figure 13: Getting the hydrophobic residues coordinates in the 1KQ0 protein.

After that, type the next commands:

```
measure center $sel2
measure minmax $sel2
```

The first one returns the geometric center of atoms in *$sel2*. And the second one returns two vectors, the first containing the minimum x , y , and z coordinates of all atoms in *$sel2*, and the second containing the corresponding maximum [1]. Figure 14 shows the result of both two commands.



```
>Main< (LabSession4) 29 %  measure center $sel2
32.14671325683594 11.422078132629395 27.802621841430664
>Main< (LabSession4) 30 % measure minmax $sel2
{9.852999687194824 -13.507440567016602 2.252041816711426} {52.930999755859375 40.1777
7633666992 47.23183059692383}
>Main< (LabSession4) 31 %
```

Figure 14: Getting structural properties, such as the geometric center or the size of a selection.

3.3 Sourcing scripts

When performing a task that requires many lines of commands, instead of typing each line in the TkConsole window, it is usually more convenient to write all the lines into a script file and load it in VMD [1].

3.3.1 Using the beta.tcl file in the 1KQ0 model

In [3], there is a simple script file called **beta.tcl**, which will be executed in VMD as an example. The script **beta.tcl** sets the colors of residues LYS and GLY to a different color from the rest of the protein by assigning them a different beta value [1].

Start a new session in VMD, and in the TkConsole window load the 1KQ0 protein (as seen in subsection Loading molecules with text commands). Then type:

```
source beta.tcl
```

The result is the same as the shown in Figure 8a. The **beta.tcl** script has the next commands, that are similar to the ones reviewed in the Obtaining and changing molecule properties with text commands subsection:

```
set crystal [atomselect top all]
$crystal set beta 0
set sel [atomselect top "resname LYS GLY"]
$sel set beta 1
$crystal delete
$sel delete
```

3.3.2 Using the alignment to principal axes in VMD script in the 1UBQ and 1KQ0 models

In [2] there is available a script that perform an alignment of a protein's principal axes with the x, y, z axes.

First, download the file **orient.tar.gz** and **la101psx.tar.gz**. The latter is a linear algebra package by Hume Integration Software [2]. Then unpack the scripts to get two folders: **la1.0** and **orient**.

Second, add the following lines to the vmd.rc file (located in C:\Program Files (x86)\University of Illinois\VMD) to make it easy to load the packages:

```
    lappend auto_path $FOLDER_PROJECT/scripts/la1.0
    lappend auto_path $FOLDER_PROJECT/scripts/orient
```

The $FOLDER_PROJECT is where all of this practice files are located.

Third, save the commands of the script in a tcl file in the $FOLDER_PROJECT, let's call it **align.tcl**:

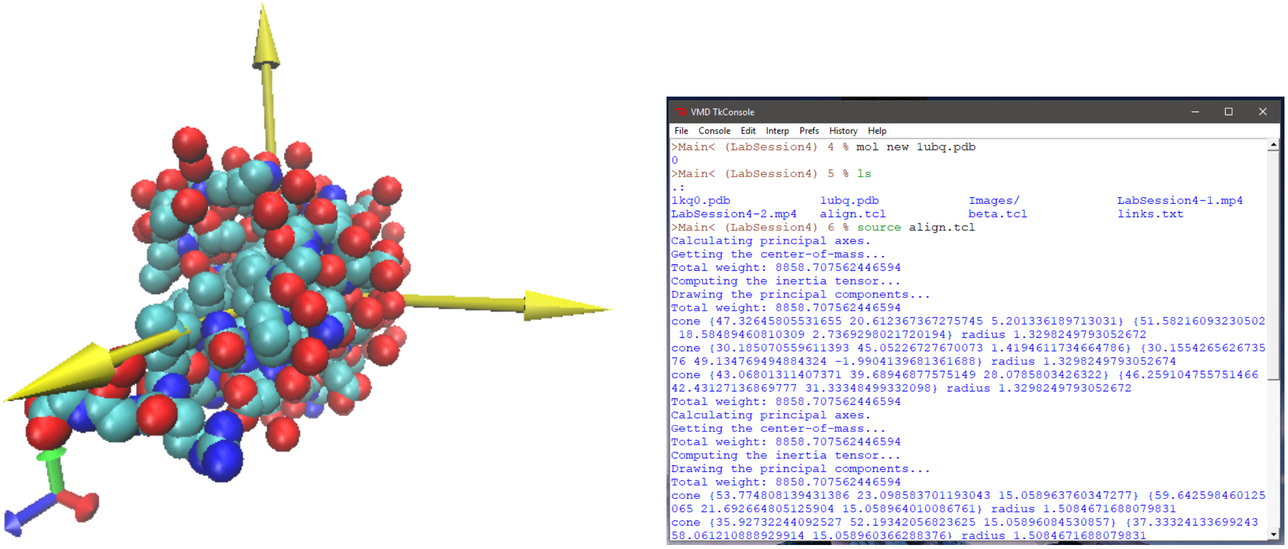
```
    package require Orient
    namespace import Orient::orient

    set sel [atomselect top "all"]
    set I [draw principalaxes $sel]
    set A [orient $sel [lindex $I 2] {0 0 1}]
    $sel move $A
    set I [draw principalaxes $sel]
    set A [orient $sel [lindex $I 1] {0 1 0}]
    $sel move $A
    set I [draw principalaxes $sel]
```

Fourth, open a new session in VMD, open the TkConsole, load the 1UBQ molecule first and load the **align.tcl** script:

```
    new molecule 1ubq.pdb
    source align.tcl
```

Figure 15 shows the resulted displayed model and the result on the TkConsole for the 1UBQ after typing these two commands.



(a) Resulted display for the 1UBQ protein model (using VDW drawing style).

```
>Main< (LabSession4) 4 % mol new 1ubq.pdb
0
>Main< (LabSession4) 5 % ls
.:
1kq0.pdb            1ubq.pdb            Images/             LabSession4-1.mp4
LabSession4-2.mp4   align.tcl           beta.tcl            links.txt
>Main< (LabSession4) 6 % source align.tcl
Calculating principal axes.
Getting the center-of-mass...
Total weight: 8858.707562446594
Computing the inertia tensor...
Drawing the principal components...
Total weight: 8858.707562446594
cone {47.32645805531655 20.612367367275745 5.201336189713031} {51.58216093230502
 18.58489460810309 2.7369298021720194} radius 1.3298249793052672
cone {30.185070559611393 45.05226727670073 1.4194611734664786} {30.1554265626735
76 49.134769494884324 -1.9904139681361688} radius 1.3298249793052674
cone {43.06801311407371 39.68946877575149 28.0785803426322} {46.259104755751466
42.43127136869777 31.33348499332098} radius 1.3298249793052672
Total weight: 8858.707562446594
Calculating principal axes.
Getting the center-of-mass...
Total weight: 8858.707562446594
Computing the inertia tensor...
Drawing the principal components...
Total weight: 8858.707562446594
cone {53.774808139431386 23.098583701193043 15.058963760347277} {59.642598460125
065 21.692664805125904 15.058964010086761} radius 1.5084671688079831
cone {35.92732244092527 52.19342056823625 15.05896084530857} {37.33324133699243
58.061210888929914 15.058960366288376} radius 1.5084671688079831
```

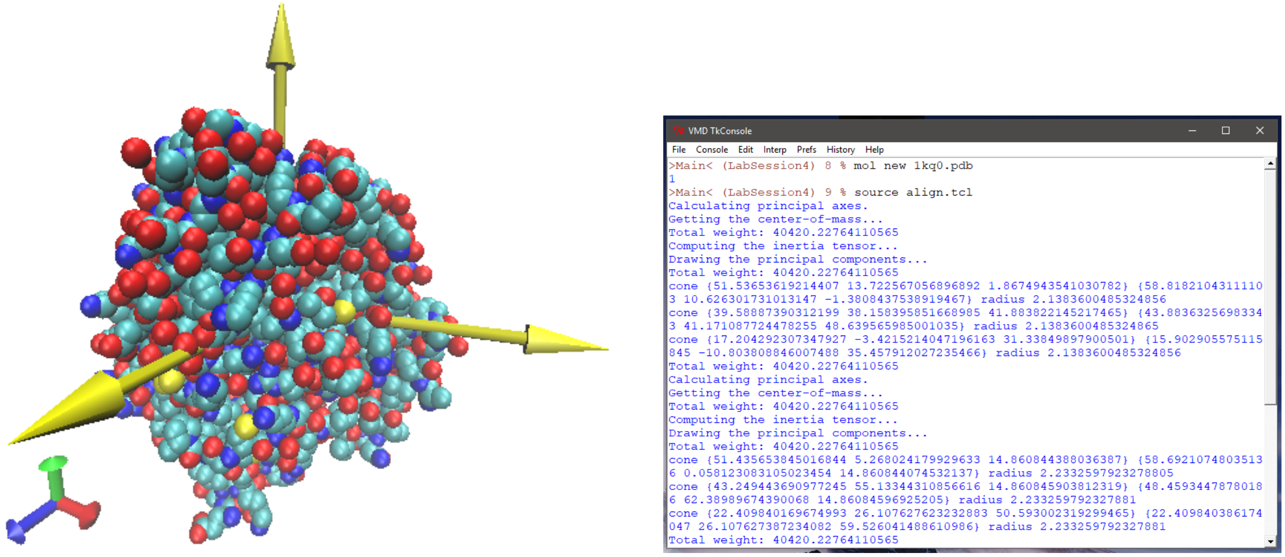
(b) Command's result after loading the **align.tcl** script in the TkConsole.

Figure 15: Applying the **align.tcl** script on the 1UBQ model.

Finally, delete the 1UBQ molecule, and repeat the same commands in the TkConsole with the 1KQ0 molecule:

```
    new molecule 1kq0.pdb
    source align.tcl
```

Figure 16 shows the resulted displayed model and the result on the TkConsole for the 1KQ0 after typing these two commands.



(a) Resulted display for the 1KQ0 protein model (using VDW drawing style).

```
>Main< (LabSession4) 8 % mol new 1kq0.pdb
1
>Main< (LabSession4) 9 % source align.tcl
Calculating principal axes.
Getting the center-of-mass...
Total weight: 40420.22764110565
Computing the inertia tensor...
Drawing the principal components...
Total weight: 40420.22764110565
cone {51.53653619214407 13.722567056896892 1.8674943541030782} {58.8182104311110
3 10.626301731013147 -1.3808437538919467} radius 2.1383600485324856
cone {39.58887390312199 38.158395851668985 41.883822145217465} {43.8836325698334
3 41.171087724478255 48.639565985001035} radius 2.1383600485324865
cone {17.204292307347927 -3.4215214047196163 31.33849897900501} {15.902905575115
845 -10.803808846007488 35.457912027235466} radius 2.1383600485324856
Total weight: 40420.22764110565
Calculating principal axes.
Getting the center-of-mass...
Total weight: 40420.22764110565
Computing the inertia tensor...
Drawing the principal components...
Total weight: 40420.22764110565
cone {51.435653845016844 5.268024179929633 14.860844388036387} {58.6921074803513
6 0.058123083105023454 14.860844074532137} radius 2.2332597923278805
cone {43.249443690977245 55.13344310856616 14.860845903812319} {48.4593447878018
6 62.38989674390068 14.86084596925205} radius 2.233259792327881
cone {22.409840169674993 26.107627623232883 50.593002319299465} {22.409840386174
047 26.107627387234082 59.526041488610986} radius 2.233259792327881
Total weight: 40420.22764110565
```

(b) Command's result after loading the **align.tcl** script in the TkConsole.

Figure 16: Applying the **align.tcl** script on the 1KQ0 model.

4 Conclusions and recommendations

Using Tcl scripts in the VMD TkConsole is a better known method for the computing area to work with the display of molecules' representations and simulations .

For an easier way to use scripting in VMD, it's highly recommended that all the Tcl commands desired for particular and specific actions should be written in a .tcl file appart, and then using the TkConsole, it can be loaded from source. This helps to develop more efficient and faster, and to have a kind of backup from the work session.

5 References

[1] Theoretical and Computational Biophysics Group, "Scripting in VMD," https://www.ks.uiuc.edu/Training/Tutorials/vmd/tutorial-html/node4.html#SECTION00042400000000000000, [Online; last access October 27, 2020].

[2] ——, "Alignment to principal axes in VMD," https://www.ks.uiuc.edu/Research/vmd/script_library/scripts/orient/, [Online; last access October 27, 2020].

[3] ——, "Index of /Training/Tutorials/vmd/vmd-tutorial-files," https://www.ks.uiuc.edu/Training/Tutorials/vmd/vmd-tutorial-files/, [Online; last access October 27, 2020].

[4] R. PDB, "1UBQ STRUCTURE OF UBIQUITIN REFINED AT 1.8 ANGSTROMS RESOLUTION." https://www.rcsb.org/structure/1UBQ, [Online; last access October 13, 2020].

[5] ——, "1KQ0 Human methionine aminopeptidase type II in complex with D-methionine," https://www.rcsb.org/structure/1KQ0, [Online; last access October 27, 2020].