THE UNIVERSITY of TEXAS

HEALTH SCIENCE CENTER AT HOUSTON

SCHOOL of HEALTH INFORMATION SCIENCES

Visualization of Biomolecular Structures

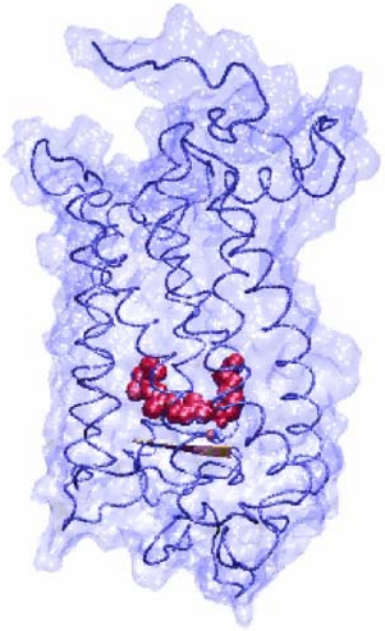
For students of HI 6327 “Biomolecular Modeling”

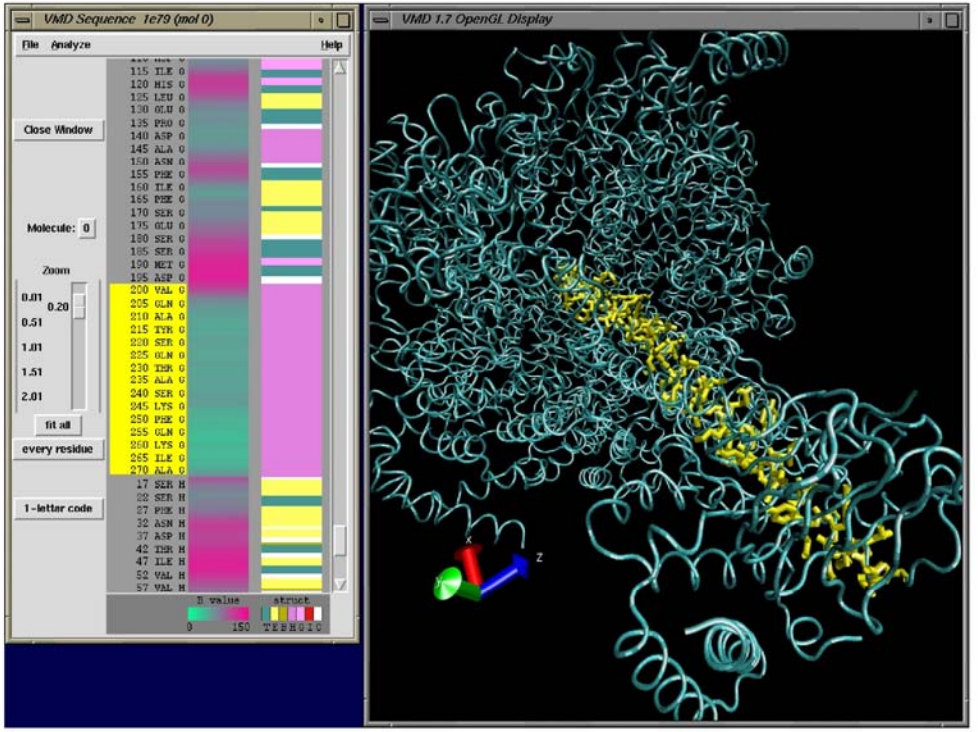
Willy Wriggers, Ph.D.

School of Health Information Sciences

http://biomachina.org/courses/modeling/03.html

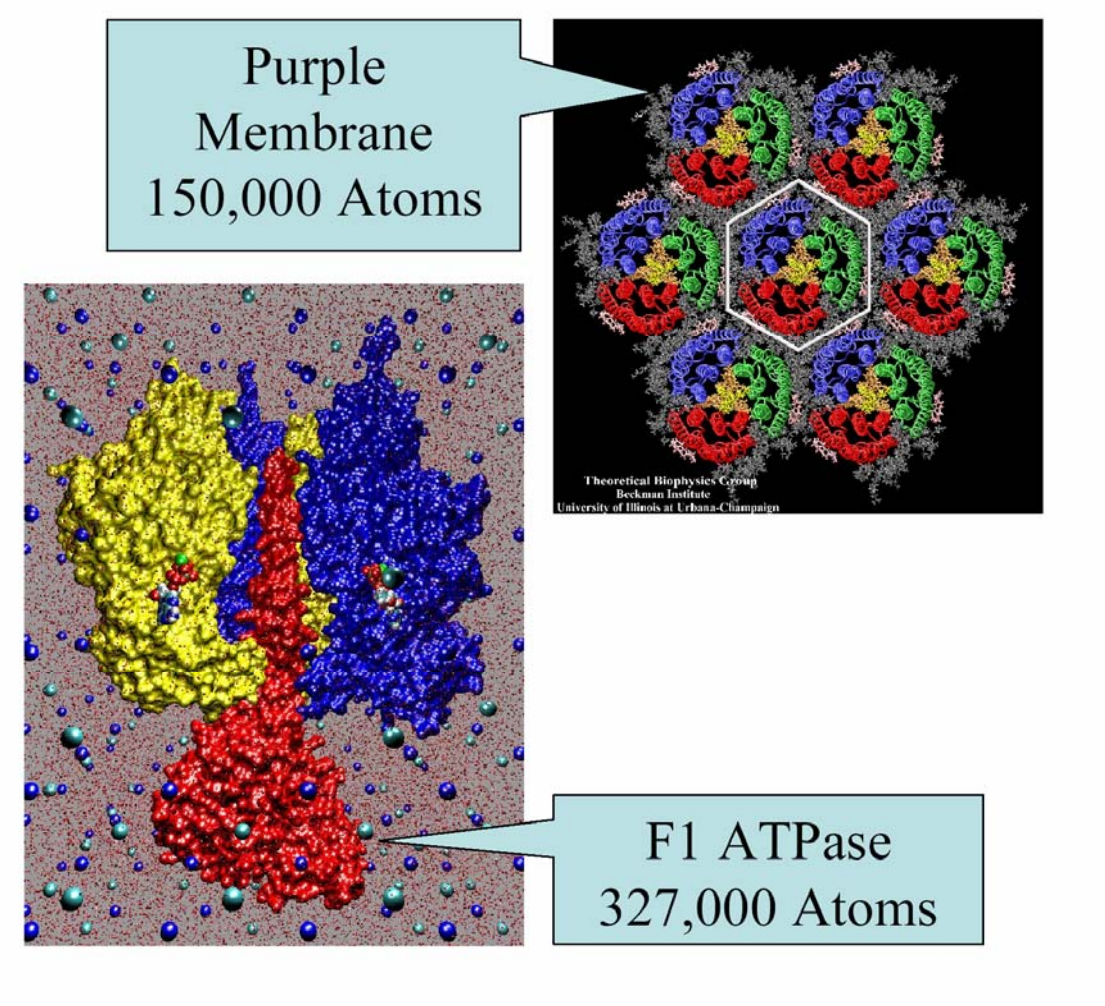
Molecular Graphics Perspectives of Protein Structure and Function





Large-Scale Structure Visualization

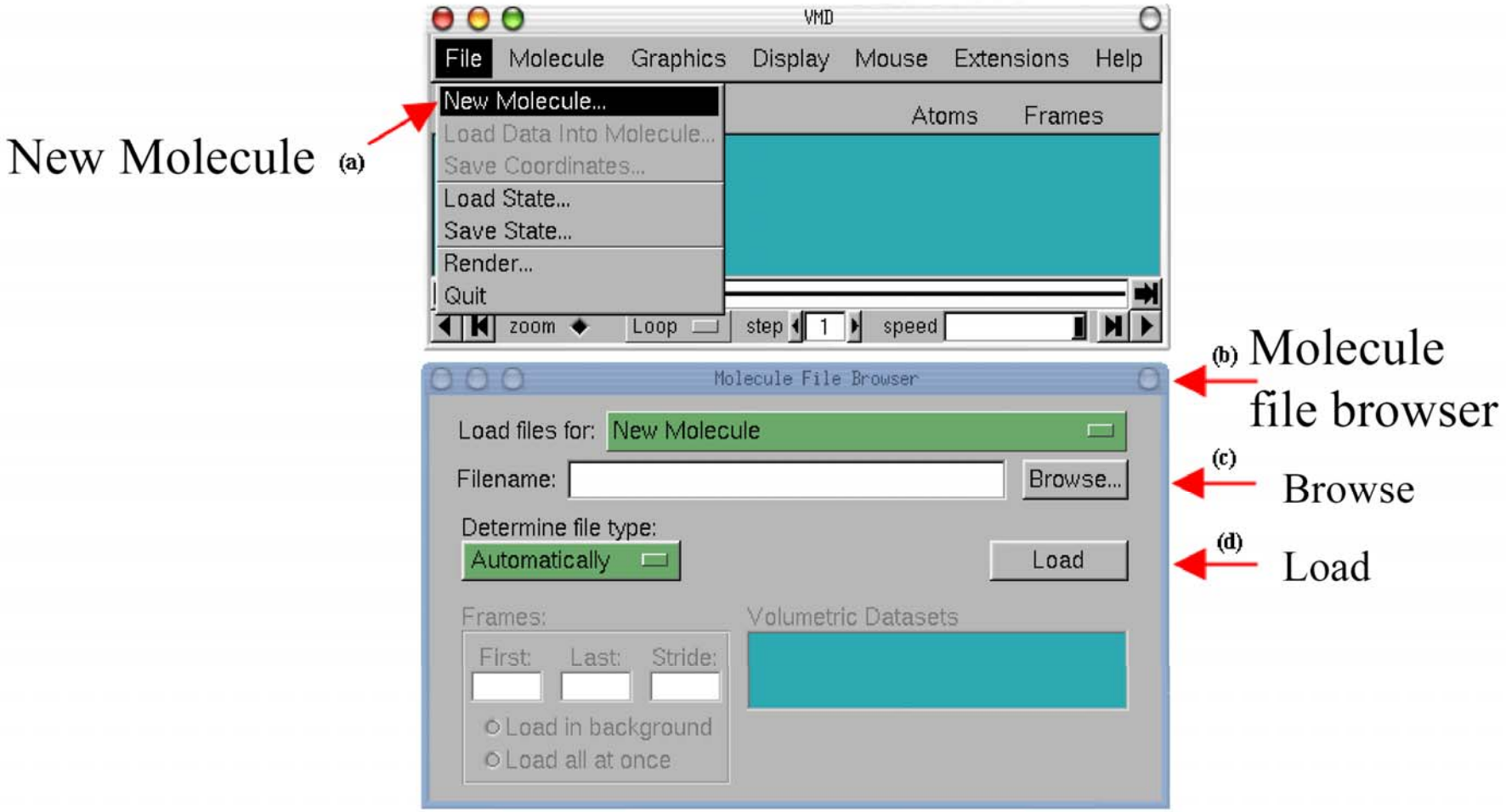
- Large structures: 300,000 atoms and up
- Complex representations
- Long trajectories: thousands of timesteps
- Volumetric data
- Molecular Dynamics: 5 ns simulation of 100K atoms produces a 12GB trajectory



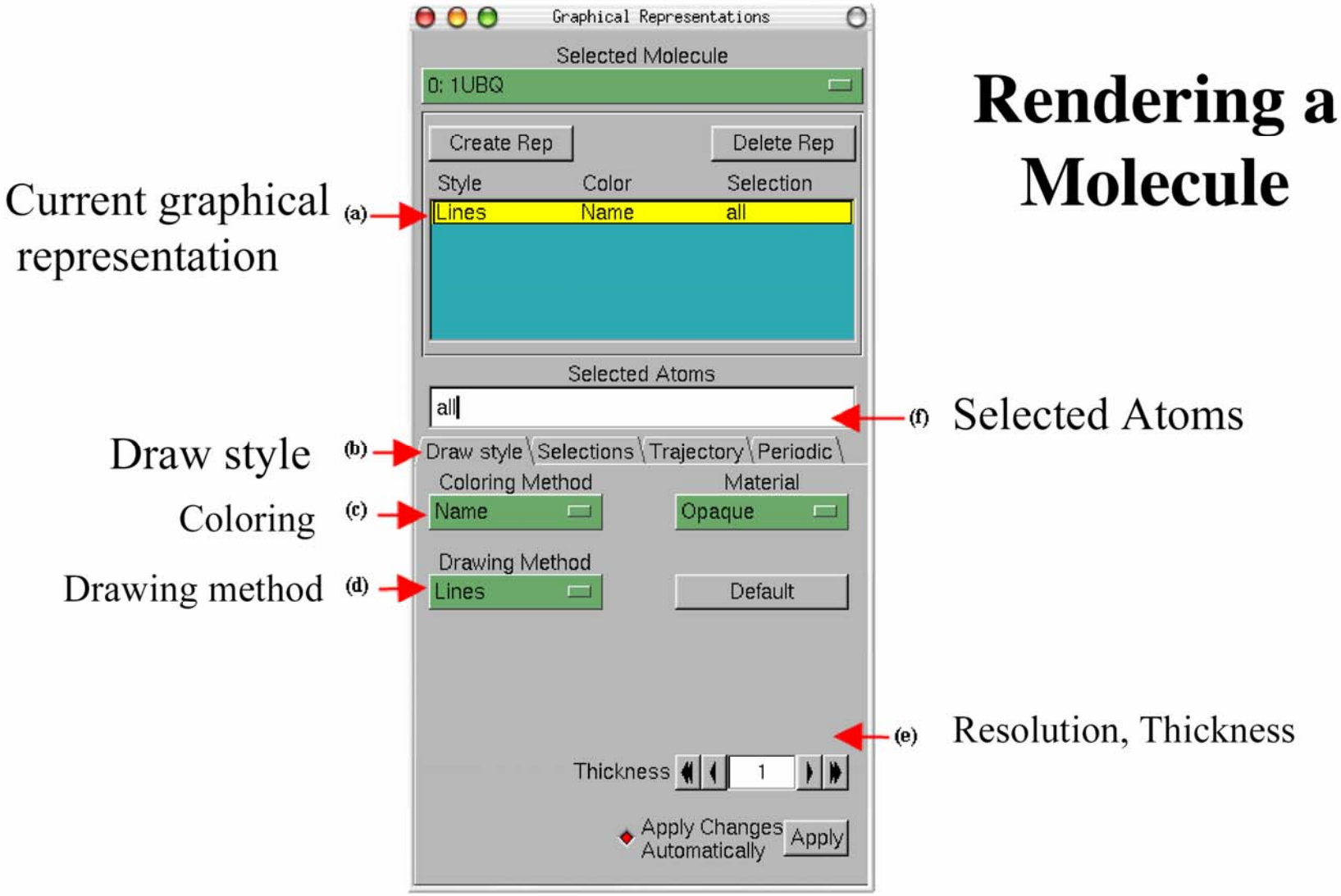
VMD Highlights

- > 30,000 registered Users
- Platforms:
  - Unix (16 builds)
  - Windows
  - MacOS X
- Display of large biomolecules and simulation trajectories
- Sequence browsing and structure highlighting
- User-extensible scripting interfaces for analysis and customization

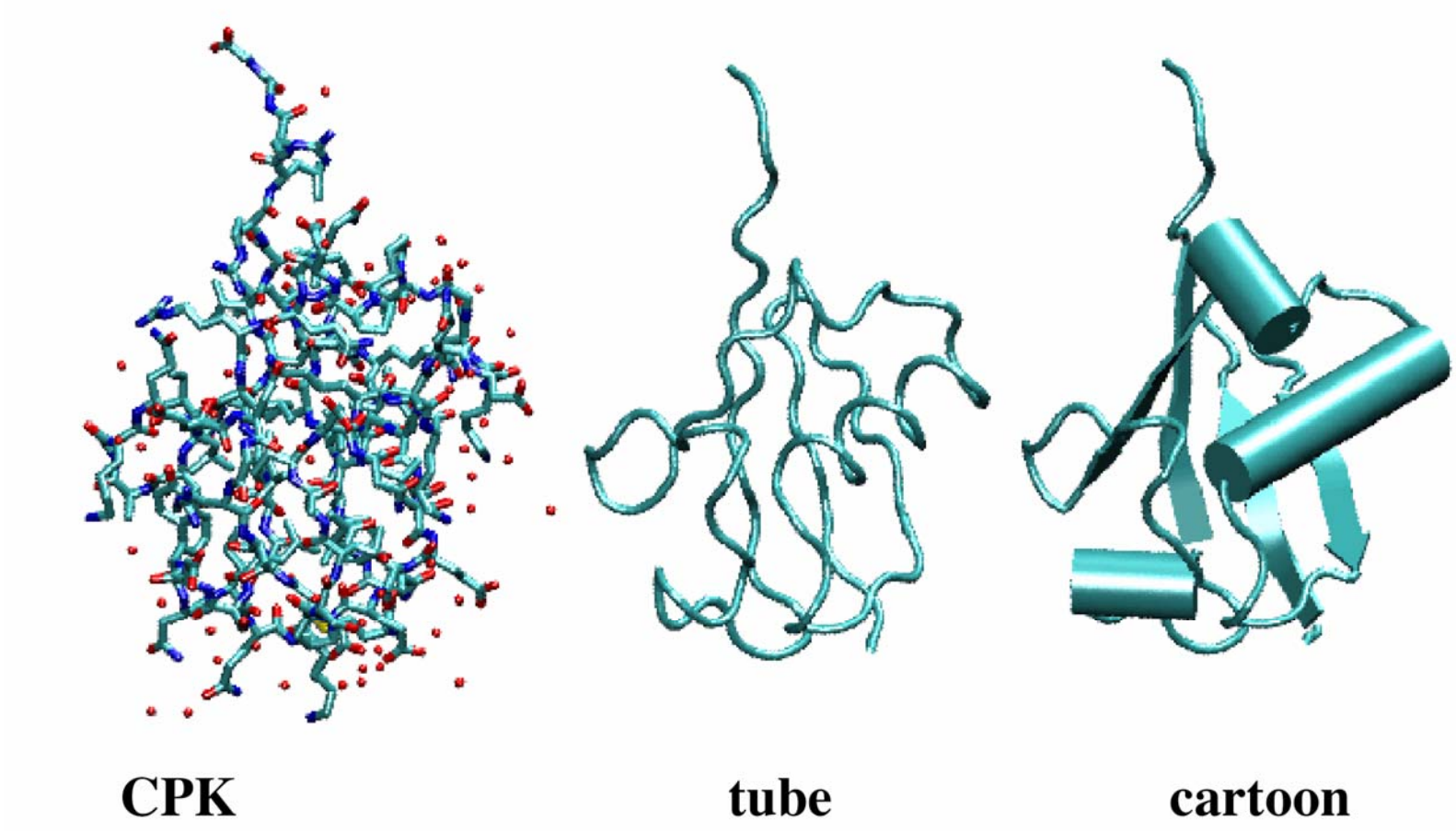
Loading a Molecule



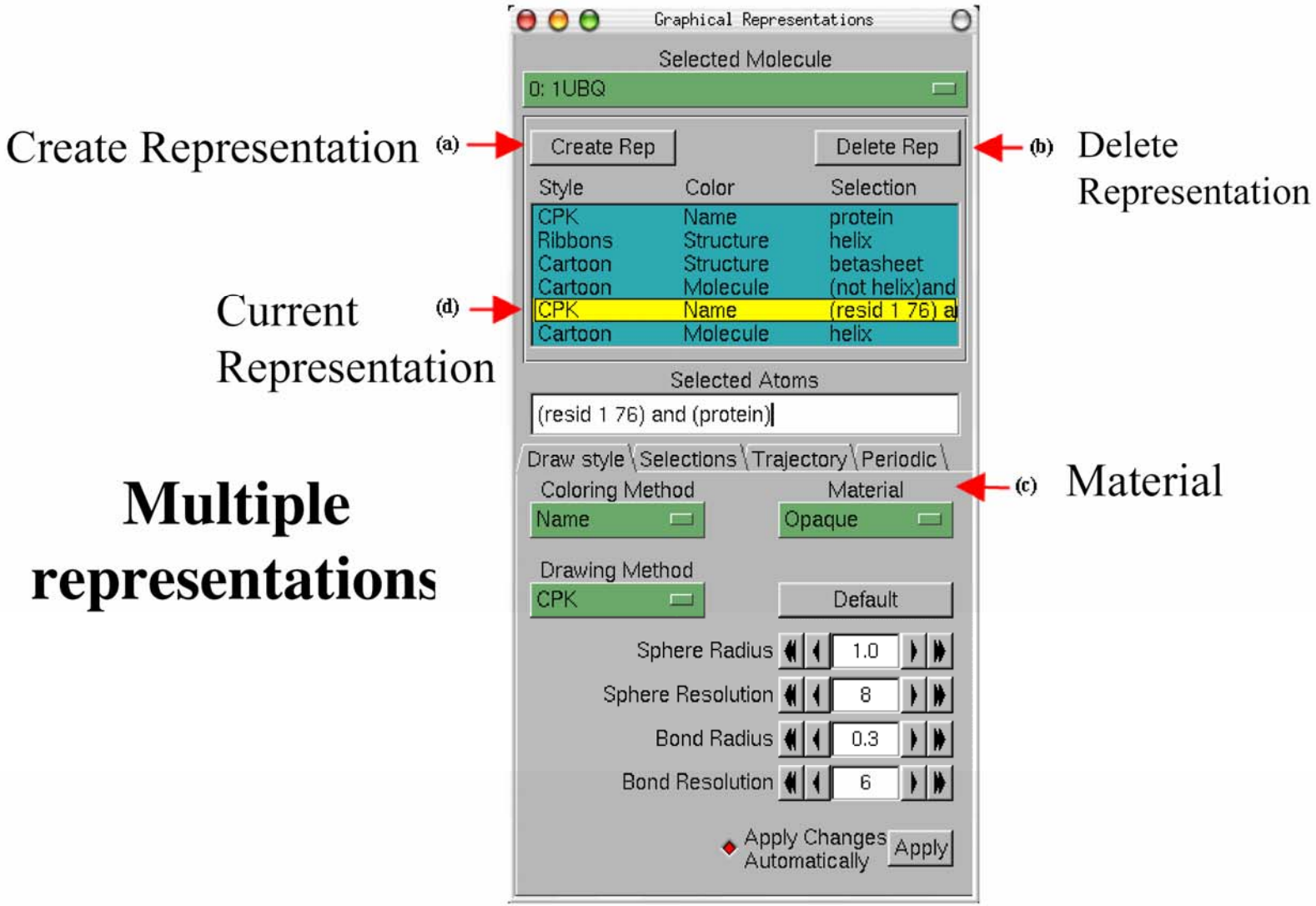
Setting Graphics Representations



Change Rendering Style



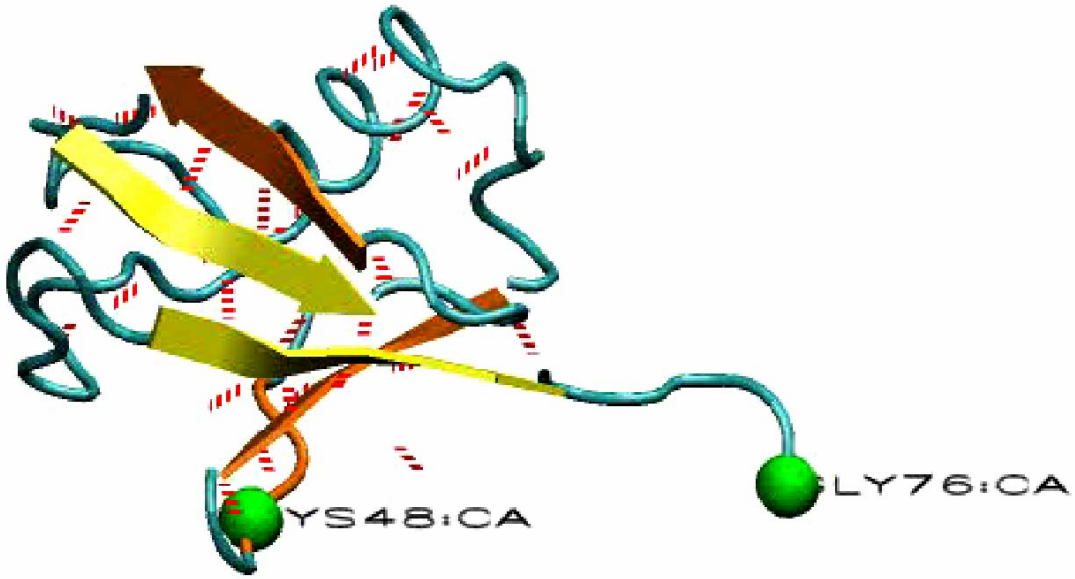
Mix Representations



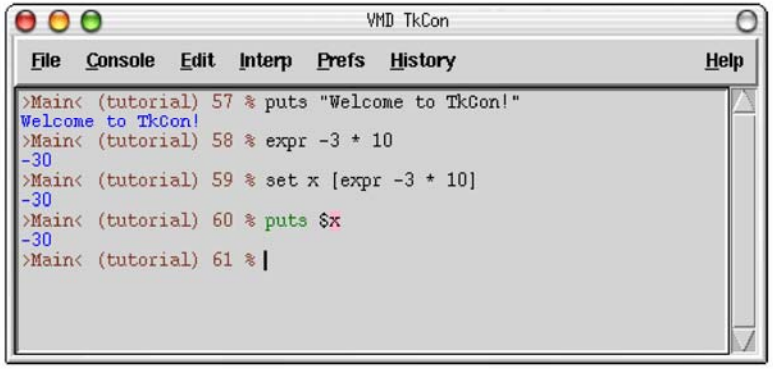
Customizing Coloring

Use VMD scripting features to color beta strands separately;

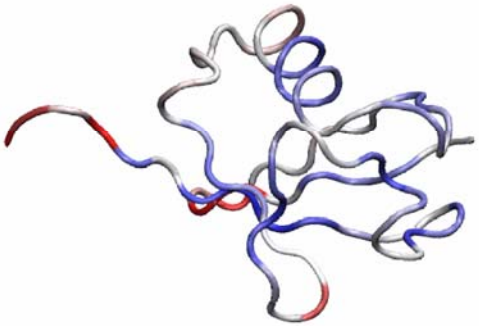
show hydrogen bonds to monitor the mechanical stability of ubiquitin

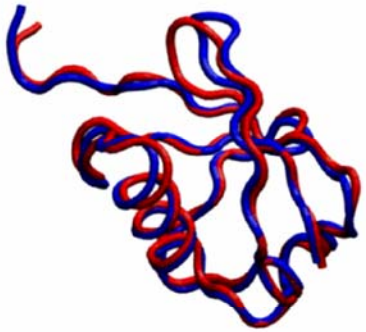


Scripting

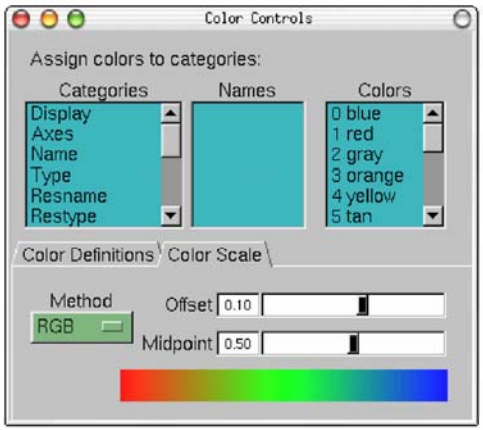


Tcl and Python supported





Left: Initial and final states of ubiquitin after spatial alignment Right (top): Color coding of deviation between initial and final



The Color Controls window showing the Color Scale tab.

Introduction to Tcl Syntax

For a scripting language, Tcl has a simple syntax:

```
cmd arg arg arg
```

A Tcl command is formed by words separated by white space. The first word is the name of the command, and the remaining words are arguments to the command.

Introduction to Tcl Syntax

```
$foo
```

The dollar sign ($) substitutes the value of a variable. In this example, the variable name is foo.

Introduction to Tcl Syntax

```
[clock seconds]
```

Square brackets execute a nested command. For example, if you want to pass the result of one command as the argument to another, you use this syntax. In this example, the nested command is clock seconds, which gives the current time in seconds.

Introduction to Tcl Syntax

```
"some stuff"
```

Double quotation marks group words as a single argument to a command. Dollar signs and square brackets are interpreted inside double quotation marks.

Introduction to Tcl Syntax

```
{some stuff}
```

Curly braces also group words into a single argument. In this case, however, elements within the braces are not interpreted.

Introduction to Tcl Syntax

\

The backslash (\) is used to quote special characters. For example, \n generates a newline. The backslash also is used to "turn off" the special meanings of the dollar sign, quotation marks, square brackets, and curly braces.

Introduction to Tcl Syntax

Below is a Tcl command that prints the current time. It uses three Tcl commands: set, clock, and puts. The set command assigns the variable. The clock command manipulates time values. The puts command prints the values.

```
set timesec [clock seconds]
puts "The time is [clock format $timesec]"
```

Note that you do not use $ when assigning to a variable. Only when you want the value do you use $. The timesec variable isn't needed in the previous example. You could print the current time with one command:

```
puts "The time is [clock format [clock seconds]]"
```

Introduction to Tcl Syntax

The Tcl syntax is used to guide the Tcl parser through three steps:

1. **Argument grouping.** Tcl needs to determine how to organize the arguments to the commands. In the simplest case, white space separates arguments. As stated earlier, the quotation marks and braces syntax is used to group multiple words into one argument. In the previous example, double quotation marks are used to group a single argument to the puts command.
2. **Result substitution.** After the arguments are grouped, Tcl performs string substitutions. Put simply, it replaces $foo with the value of the variable foo, and it replaces bracketed commands with their result. That substitutions are done *after* grouping is crucial. This sequence ensures that unusual values do not complicate the structure of commands.
3. **Command dispatch.** After substitution, Tcl uses the command name as a key into a dispatch table. It calls the C procedure identified in the table, and the C procedure implements the command. You also can write command procedures in Tcl. There are simple conventions about argument passing and handling errors.

Introduction to Tcl Syntax

Here is another example:

```
set i 0
while {$i < 10} {
      puts "$i squared = [expr {$i*$i}]"
      incr i
}
```

Here, curly braces are used to group arguments without doing any substitutions. The Tcl parser knows nothing special about the while command. It treats it like any other command. It is the implementation of the while command knows that the first argument is an expression, and the second argument is more Tcl commands. The braces group two arguments: the boolean expression that controls the loop and the commands in the loop body. We also see two math expressions: the boolean comparison and multiplication. The while command automatically evaluates its first argument as an expression. In other cases you must explicitly use the expr command to perform math evaluation.

Introduction to Tcl Syntax

It is very easy to write command procedures. They can do everything from accessing databases to creating graphical user interfaces. Tcl, the language, doesn't really know what the commands do. It just groups arguments, substitutes results, and dispatches commands.

One Last Example:

```
proc fac {x} {
      if {$x < 0} {
            error "Invalid argument $x: must be pos."
      } elseif {$x <= 1} {
            return 1
      } else {
            return [expr {$x * [fac [expr {$x-1}]]}]
      }
}
```

Tcl Resources

WWW:
http://www.tcl.tk/man/tcl8.5/tutorial/tcltutorial.html

Books:
Welch: Practical Programming in Tcl and Tk (Prentice Hall)
Ousterhout: Tcl and the Tk Toolkit (Addison Wesley)

Any Questions?

…next : VMD Tutorial