PMV and Underlying Components

Michel F. Sanner

The Molecular Graphics Laboratory

The Scripps Research Institute
La Jolla, California

SciPy’02
CalTech Pasadena CA. Sept. 2002
SCHEDULE

- Reusable Components
  - MolKit, DejaVu, ViewerFramework, ...
- From building blocks to applications
  - PMV: a molecular visualization program
- Extending PMV
  - ADT: setting up and running AutoDock
- Conclusion
Re-usable components

- MolKit:
  - read/write/represent/manipulate molecules
- Mslib:
  - molecular surface calculation
- DejaVu:
  - General purpose 3D geometry viewer
- ViewerFramework:
  - Visualization application template
- Mslib, PyBabel, PyMead, SFF, Isocontour, ...

(Sophie I. Coon, Michel F. Sanner and Art J. Olson, Re-usable components for structural bioinformatic, 9th Python Conference 2001)
TreeNode

- parent
- top
- children
- elementType
- name
- adopt(child)

TreeNode

TreeNodeSet(ListSet)

- [TreeNode1, TreeNode2, ...]
- __getattr__(self, name)

returns

[ TreeNode1.name, TreeNode2.name, ...]
TreeNode and TreeNodeSet specialization

TreeNode

- Molecule
  - Residue
  - Chain
  - Protein
  - ...
- Atom
- SecondaryStructure
  - Helix
  - Strand
  - Turn
  - Coil
  - ...

TreeNodeSet

- MoleculeSet
  - ResidueSet
  - ChainSet
  - ProteinSet
  - ...
- AtomSet
- SecondaryStructureSet
  - HelixSet
  - StrandSet
  - TurnSet
  - CoilSet
  - ...

SciPy'02, © M.Sanner, TSRI
from MolKit.pdbParser import PdbParser
parser = PdbParser('1crn.pdb')
mols = parser.parse()
Examples

```python
>>> from MolKit import Read
>>> molecules = Read('./1crn.pdb') # Read returns a ProteinSet
>>> mol = molecules[0]
>>> print mol.chains.residues.name
>>> print mol.chains.residues.atoms[20:85].full_name()
>>> from MolKit.molecule import Atom
>>> allAtoms = mol.findType(Atom)
>>> set1 = allAtoms.get(lambda x: x.temperatureFactor >20)
>>> allResidues = allAtoms.parent.uniq()
>>> import Numeric
>>> for r in allResidues:
...     coords = r.atoms.coords
...     r.geomCenter = Numeric.sum(coords) / len(coords)
```
from DejaVu import Viewer
vi = Viewer()

from DejaVu.Spheres import Spheres
centers = [[0,0,0],[3,0,0],[0,3,0]]
s = Spheres('sph', centers = centers)
s.Set(quality=10)
vi.AddObject(s)
Features

- OpenGL Lighting and Material model
- Arbitrary clipping planes
- Multiple light sources
- Material editor
- DepthCueing (fog), global anti-aliasing
- glScissors/magic lens
- Object hierarchy with transformation and rendering properties inheritance
- Multi-level picking
- Extensible set of geometries
Geometries

Geom

- PolyLine
- Points
- Spheres
- Labels
- Arc3D...

IndexedGeoms

- IndexedPolyLines
- IndexedPolygons
- TriangleStrip
- QuadStrip
- Cylinders ...
Mslib

- Python wrapper of MSMS
  - $\text{XYZR} \rightarrow \text{reduced Surface}$
    - $\rightarrow \text{Analytical SES}$
    - $\rightarrow \text{triangulated SES}$
- Surface genus, areas
- buried surface calculation
Design features

- Dynamic loading of commands
- Python shell for scripting
- Dual interaction mode (GUI/Shell)
- Support for command:
  - development, logging, GUI, dependencies
- Lightweight commands: Macros
- Dynamic commands (introspection)
- Extensible set of commands

Access to documentation
PMV: From Building Blocks to applications
PMV Capabilities

- Display/compute:
  - Lines, CPK, Stick and Balls, Surfaces, Splines, Ribbons, H-bonds, bond-order, Gasteiger and Kollmann charges

- Editing:
  - deleting atoms, adding hydrogens

- Electrostatic potential
  - MEAD*, APBS*

- Molecular Dynamics:
  - Amber*
Automated docking of a flexible ligand to macromolecules using affinity grids
AutoDock ToolKit (ADT)

- **AutoTors**: ligand preparation
- **AutoGpf**: grid definition
- **AutoDpf**: docking parameters definition
- **AutoStart**: job launching and monitoring
- **AutoAnalyze**: docking results analysis
ADT: extending PMV
ADT: Architecture

ADT
- AutoTors, AutoGpf
- AutoDpf, AutoLaunch
- AutoAnalyze

PMV
- ViewerFramework
- DejaVu

Python Interpreter
- Msms Commands
- Mslib
- MolKit
- Idle
- Numeric
- PyOpenGL
- Tkinter

SciPy’02, © M.Sanner, TSRI
Conclusion

- Flexible software built from components
- Take advantage of Python’s advanced features
- Molecular manipulation environment
Acknowledgments

- Molecular Graphics Lab. at TSRI
- NIH (NBCR RR08605 and ITR EIA 0121282)
- NSF (NPACI CA ACI9619020)
- Swiss National Science Foundation
- Daniel Stoffler
- Sophie Coon
- Ruth Huey

Available at:
http://www.scripps.edu/~sanner/python