

Web-based 3D molecular viewers: an update

Before:

- Only desktop applications could make use of GPU-based hardware acceleration through OpenGL (Chimera, VMD, PyMol etc...)
- Web browsers couldn't display 3D content without plug-ins
- Development of java applets-based viewers:
 - Jmol
 - OpenAstex Viewer

Recently:

- Java install base is shrinking due to heavily publicized java security failures
- User is annoyed with multiple security prompt
- Java applets are no longer supported in Chrome
- Development of JavaScript-based viewers:
 - Jmol → java2script (java to javascript converter) → Jsmol:
 - Same as Jmol but very slow
 - Old fashioned

Now:

- Modern browsers can use hardware-accelerated graphics through the WebGL standard :
 - Firefox since version 5 (2010)
 - Chrome since version 9 (2011)
 - Safari since version 5.1 (2011)
 - Opera since version 12 (2012)
 - Internet Explorer since version 11 (2013)
- GLMol is the first Web-based 3D molecule viewer to make use of WebGL
 - Relies on three.js library
 - Still relatively slow
- New WebGL viewers start to appear since 2013 :
 - PV
 - NGL
 - 3DMol.js
 - Molmil
 - Jolecule

RPBS

Ressource Parisienne en Bioinformatique Structurale.

Usage:

	Citations	Notable sites using it
NGL	32	RCSB(PDB)
3Dmol.js	27	CATH
Molmil	4	PDBj
PV	N/A	RCSB(PDB) SWISS-MODEL
Jolecule	N/A	?

RPBS

Ressource Parisienne en Bioinformatique Structurale.

Features:

	PV	NGL	3DMol.js	MolMil
File supports	pdb, sdf	pdb, sdf, mol2, pqr, gro, cif, mmtf	pdb, sdf, mol2, xyz	pdb, cif, gro, trr, xtc, cor, mpbf, ccp4,mdl, mol2
Molecular surface		✓	✓	✓
Sphere, stick, line, trace, cartoon etc...	✓	✓	✓	✓
Clickable interactivity	✓	✓	✓	✓
Geometric shapes	✓	✓	✓	
Hosted viewer		✓	✓	✓

Performance comparison:

	JMol	JSMol	GLMol	PV	NGL	3Dmol.js
Construct	0.136s	0.874s	6.372s	2.146	1.095s	0.776s
Rotate	0.053s	0.207s	0.673s	0.001s	0.002s	0.002s

Time to load 3M8L (capsid protein) with 12,375 atoms and then rotate it.

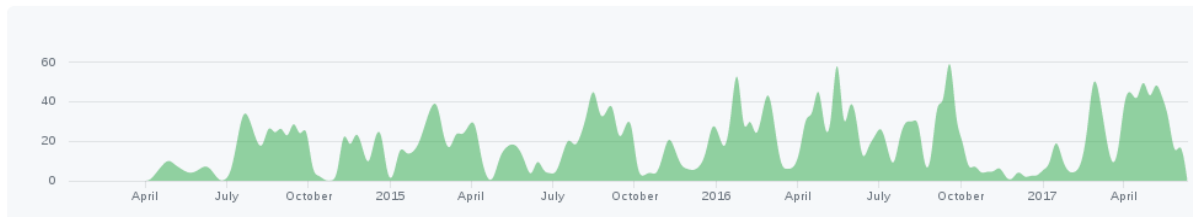
Conclusion : 3Dmol.js may be a bit faster at building large molecules. But once loaded, PV, NGL and 3Dmol.js performances are rather similar.

RPBS

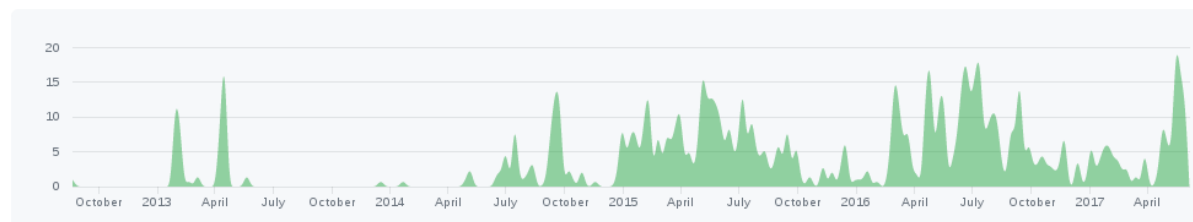
Ressource Parisienne en Bioinformatique Structurale.

Development activity:

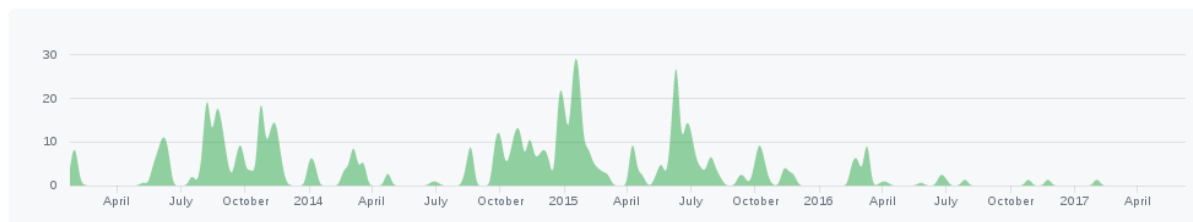
NGL



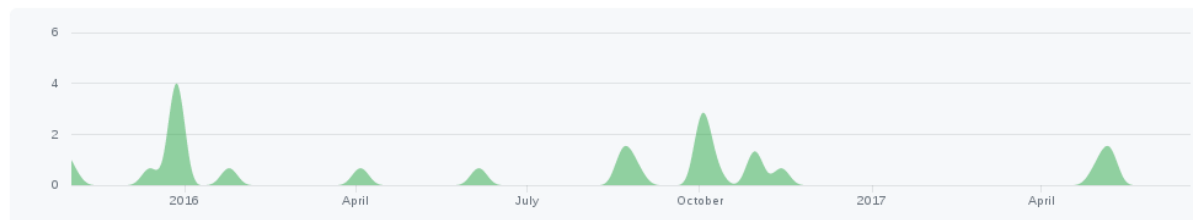
3Dmol.js



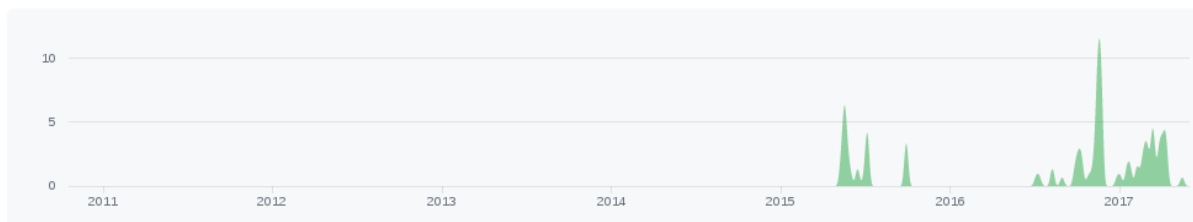
PV



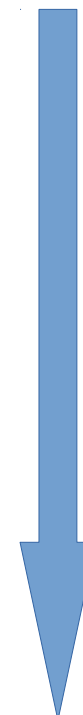
Molmil



Jolecule



+



-

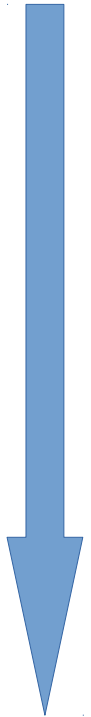
RPBS

Ressource Parisienne en Bioinformatique Structurale.

API usage:

	deployment speed	documentation	examples
NGL	++	++	+++
3Dmol.js	+++	++	+
PV	++	+++	-
Molmil	?	+	-

+



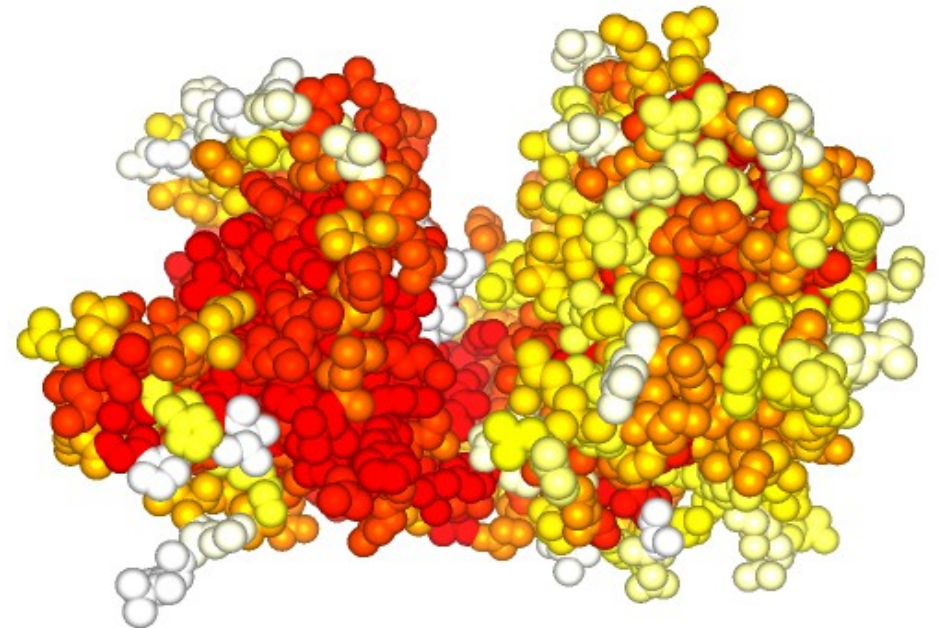
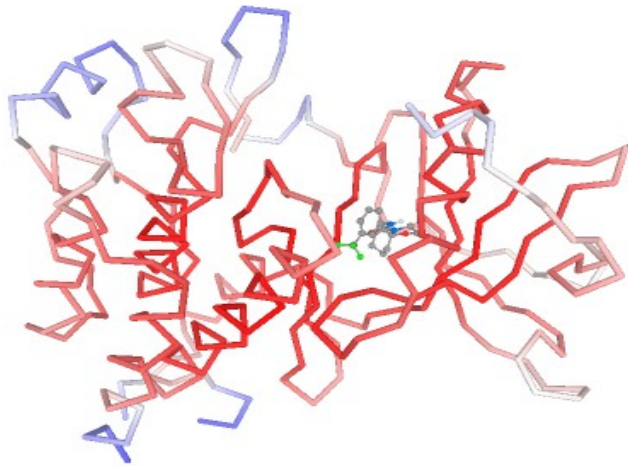
-

RPBS

Ressource Parisienne en Bioinformatique Structurale.

PV:

- <https://github.com/biasmv/pv>
- Developed by Marco Biasini (+ 12 other contributors)
- MIT license
- Very light (156 K)
- Beautiful renderings
- Oriented towards proteins only
- Embedded viewer only (no hosted viewer)
- No longer developed (last commit : Feb 23rd 2017)
- Lacked some features at the beginning like multi-model pdb files management, coloration by b-factor, etc.
- Still lacks surfaces rendering



PV:

Installation:

```
git clone https://github.com/biasmv/pv
```

Invocation:

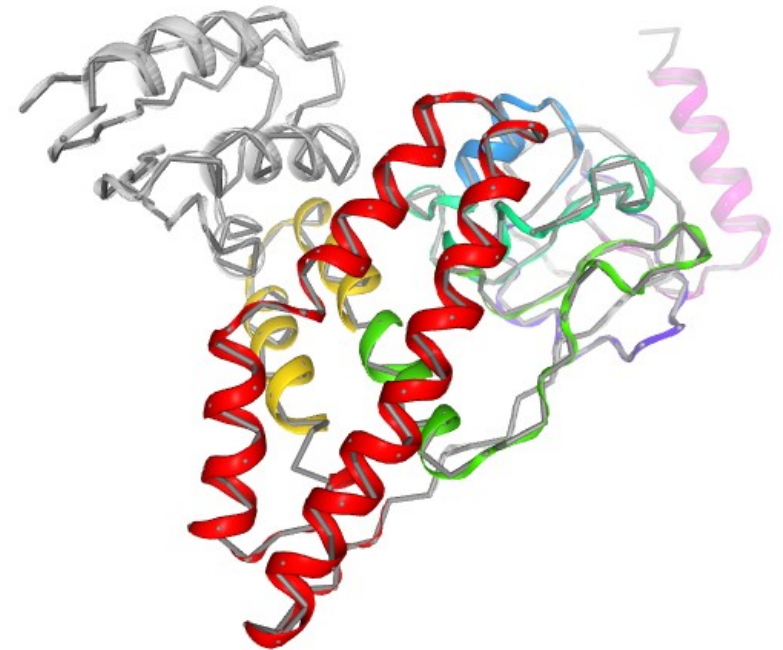
```
<body>
  <div id="viewport"></div>
</body>

<script src='jquery.js'></script>
<script src='pv.min.js'></script>

<script>
  var pv = pv.Viewer(document.getElementById("viewport"), {
    width: 658,
    height: 616,
    antialias: true,
    quality: 'high',
    outline: false
  });
  var structure;
  $(document).ready(function() {
    $.ajax({ url : file, success : function(data) {
      structure = io.pdb(data);
      pv.cartoon("cartoon", structure, { color: color.ssSuccession() });
    }}) ;
  });
</script>
```

NGL:

- <https://github.com/arose/ngl>
- Developed by AS. Rose and P.W. Hildebrand (Charité–Universitätsmedizin Berlin, Germany)
- MIT license
- Can read both proteins and RNA/DNA molecules
- Very well documented, gallery with lots of examples
- Very complete
- Embedded viewer as well as hosted viewer :
<http://proteininformatics.charite.de/ngl-tools/ngl/html/ngl.htr>
- Maybe not as pretty as PV (more aliasing)



NGL:

Installation:

```
git clone https://github.com/arose/ngl
```

Invocation:

```
<body>
  <div id="viewport"></div>
</body>

<script src='jquery.js'></script>
<script src='ngl.js'></script>

<script type="text/javascript">
  var stage = new NGL.Stage("viewport");
  stage.viewer.setBackground("white");
  $(document).ready(function() {
    stage.loadFile(data).then( function (o) {
      o.addRepresentation("cartoon", { color: "atomindex" });
      o.autoView();
    });
  });
</script>
```

3Dmol.js:

- <https://github.com/3dmol/3Dmol.js>
- Developed by N. Rego and D. Koes (University of Pittsburgh, USA)
- BSD licence
- Looks very promising (not tested yet!)
- Hosted viewer at <http://3dmol.csb.pitt.edu/viewer.html>
- Can be used in 3 ways :

JavaScript API

```
$3Dmol.download('pdb:3M8L', viewer);  
viewer.setStyle({chain:'A'}, {sphere:({}) });  
viewer.setStyle({chain:'B'},  
  {cartoon:{color: 'spectrum'} });  
viewer.setStyle({chain:'C'}, {stick:({}) });  
viewer.render();
```

HTML

```
<div style="height: 600px; width: 600px;"  
  class='viewer_3Dmoljs' data-pdb='3M8L'  
  data-backgroundcolor='0xffffffff'  
  data-select1='chain:A'  
  data-style1='sphere'  
  data-select2='chain:B'  
  data-style2='cartoon:color=spectrum'  
  data-select3='chain:C'  
  data-style3='stick'></div>
```

Hosted viewer

```
http://3dmol.csb.pitt.edu/viewer.html?  
pdb=3M8L&select=chain:A&style=sphere&  
select=chain:B&style=cartoon:color=spectrum&  
select=chain:C&style=stick
```

3Dmol.js:

Installation:

Can be invoked directly from <http://3Dmol.csb.pitt.edu/build/3Dmol-min.js>

Or downloaded:

```
git clone https://github.com/dkoes/3Dmol.js.git
```

Invocation:

```
<body>
  <div id="viewport"></div>
</body>

<script src="http://3Dmol.csb.pitt.edu/build/3Dmol-min.js"></script>

<script type="text/javascript">
  var viewer = $3Dmol.createViewer($("#div"));
  $3Dmol.download("pdb:1M08",viewer,{multimodel:true, frames:true},function(){
    viewer.setStyle({}, {cartoon:{color:"spectrum"}});
    viewer.render();
    viewer.addSurface($3Dmol.SurfaceType.VDW, {
      opacity:0.85,
      voldata: new $3Dmol.VolumeData(volumedata, "cube"),
      volscheme: new $3Dmol.Gradient.ROYGB(range[1],range[0])
    },{});
  });
</script>
```

Integration into Mobylye / Galaxy

Mobylye

- Type-specific visualizations (applets)
- Service-specific visualizations

Galaxy

- Type-specific visualizations

Integration into Mobylye:

Type-specific visualizations are set up with XML files

/opt/mobylye/core/Local/Services/Viewers/pv.xml

```
<interface type="viewer">
  <center xmlns="http://www.w3.org/1999/xhtml" style="height: 100%">
    <applet codebase="/data/services/servers/local/viewers/jmol" code="JmolApplet" archive="JmolApplet.jar">
      <param data-parametername="structfile" name="load" value="data-url"/>
      <param name="progressBar" value="true"/>
    </applet>
  </center>
</interface>
</head>
<parameters>
  <parameter>
    <name>structfile</name>
    <prompt>3D Structure</prompt>
    <type>
      <datatype>
        <superclass>AbstractText</superclass>
        <class>_3DStructure</class>
      </datatype>
      <dataFormat>PDB</dataFormat>
      <dataFormat>mol2</dataFormat>
      <dataFormat>sdf</dataFormat>
    </type>
  </parameter>
</parameters>
```


Integration into Mobylye:

Mobylye was designed to integrate java-applet viewers only:
can be overcome with the use of iframes

/opt/mobylye/core/Local/Services/Viewers/pv.xml

```
<parameters>
  <parameter>
    <name>structfile</name>
    <prompt>3D Structure</prompt>
    <type>
      <datatype>
        <superclass>AbstractText</superclass>
        <class>_3DStructure</class>
      </datatype>
      <dataFormat>PDB</dataFormat>
    </type>
    <interface type="viewer">
      <center xmlns="http://www.w3.org/1999/xhtml" style="height: 100%">
        <iframe frameborder="0" id="pv_frame" src="/portal/js/pv/viewer.html"/>
          <param data-parametername="structfile" name="load" value="data-url"/>
        </center>
      </interface>
    </parameter>
  </parameters>
```

Integration into Mobylye:

Glimpse of JavaScript code which is integrated into the iframe

/opt/mobylye/htdocs/portal/js/pv/viewer.html

```
function getURLParameter(name) {
  return decodeURIComponent((new RegExp('[?]&' + name + '=' + '([^&]+?)(&#;|$)').exec(document.referrer))||[, ''])...
}

$(document).ready(function() {
  var structfile;
  var param_structfile = getURLParameter("parameter");
  if (!param_structfile) {
    structfile = $(window.frameElement).parent().find('param').prop('value');
  } else {
    structfile = param_structfile.split("|")[1];
  }
  loadModel(structfile, "protein", cartoon);
});
```

Integration into Mobylye:

Type-specific visualizations

Input data

PDB file (_3DStructure)

basicbuilder.pdb (PDB)

HEADER	BuiltByPDBMac				
ATOM	1	N	LEU	1	-1.013 2.446 -3.787
ATOM	2	CA	LEU	1	-2.121 1.764 -3.102
ATOM	3	C	LEU	1	-1.569 0.639 -2.225
ATOM	4	O	LEU	1	-1.758 -0.551 -2.521
ATOM	5	CB	LEU	1	-3.084 1.172 -4.147
ATOM	6	CG	LEU	1	-3.711 2.249 -5.019
ATOM	7	CD1	LEU	1	-4.466 1.658 -6.198
ATOM	8	CD2	LEU	1	-4.695 3.109 -4.242

full screen pv openastex ngl jmol bookmark as basicbuilder.j or

PV - JavaScript Protein Viewer 1.3.1
3D structure viewer

Input style

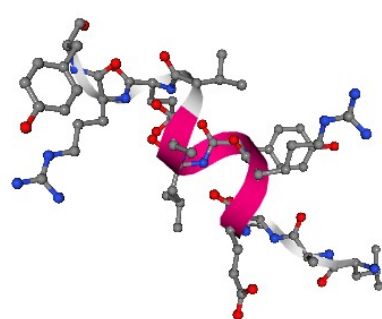
- Cartoon
- Balls & Sticks
- Lines
- Line Trace
- Smooth Line Trace

Color by

- Uniform
- Succession
- Occupancy

<http://biasmv.github.io/pv/index.html>
Author(s): Marco Biasini

NGL Viewer 0.5
3D structure viewer



- backbone
- ball+stick
- base
- cartoon
- contact
- crossing
- helixorient
- hyperball
- label
- licorice
- line
- point
- ribbon
- rocket
- rope
- spacefill
- surface
- trace
- tube

http://proteininformatics.charite.de/ngl/doc/index.html#User_manual/Introduction/Welcome
Author(s):

Integration into Mobylye:

Service-specific visualizations are set up with Jinja templates

/service/templates/pepfold3.vis.html

```
$(document).ready(function() {  
  {% if receptor is defined %}  
    loadModel('{{ wdir }}/{{ receptor }}', "receptor");  
  {% endif %}  
  var q = queryString.parse(location.search);  
  if (q.model == null) {  
    q.model = $("#slider option:first").attr('value');  
  }  
  
  $("#slider select").change(function () {  
    var model_file = $(this).find(":selected").attr('value');  
    loadModel('{{ wdir }}/' + model_file, "protein");  
    q.model = model_file;  
    $("#bookmark").attr('href', location.origin + location.pathname + '?' + queryString.stringify(q));  
  });  
  $("#slider select").val(q.model).change();  
  $("#{{ color }}").prop("checked", true);  
});
```


Integration into Mobylye:

results

Output

Visualize the best models (HTML)

bestModels.html



Input style

- Cartoon
- Balls & Sticks
- Lines
- Line Trace
- Smooth Line Trace
- Trace

Color by

- Chain
- Uniform
- Rainbow

Model

Model 1

Information

Selected

[Bookmark this page](#)

Note

You can rotate the model by dragging with the left mouse button and zoom by scrolling. You can display the residue number by left-clicking on the molecule.

full screen [bookmark](#) as bestModels or

Visualize the best models (HTML)

clusters.html

Download archive of all models [here](#)

#	file	r_BCscoreavg	gdt	max	q	tm	sOPEP
# cluster 1 - Size: 4							
	2e2l-modell	0.999	0.560	0.658	0.595	0.488	0.497-78.1554
	2e2l-modell_1	0.930	0.565	0.668	0.595	0.481	0.514-76.4861
	2e2l-modell_2	0.982	0.584	0.682	0.617	0.499	0.537-76.3501

Service-specific visualizations:

- Each service can produce its own visualization (html page)
- The result page aggregates all results
- The result page can be bookmarked

RPBS

Ressource Parisienne en Bioinformatique Structurale.

Live example:

<http://mobylye.rpbs.univ-paris-diderot.fr/cgi-bin/portal.py#jobs::InterEvDock.O30995888458967>

<http://mobylye.rpbs.univ-paris-diderot.fr/data/jobs/MTiOpenScreen/H07673856726885/output.vis.html>

Integration into Galaxy:

Type-specific viewers are set up with XML config file and mako template

/opt/galaxy/config/plugins/visualizations/pv/templates/pv.mako

```
<script type="text/javascript">
  var hdald = '${trans.security.encode_id( hda.id )}',
      hdaExt = '${hda.ext}',
      ajaxUrl = "${h.url_for( controller='/datasets', action='index')}" + hdald + "/display?to_ext=" + hdaExt;

  var viewer = pv.Viewer(document.getElementById('gl'), {
    width: 600,
    height: 600,
    antialias: true,
    quality: 'medium',
    outline: false
  });

  ...
  $(document).ready(function() {
    $("#color input option:first").prop('checked', true);
    loadModel(ajaxUrl, 'main', cartoon);
  });
</script>
```

RPBS

Ressource Parisienne en Bioinformatique Structurale.

Integration into Galaxy:

Type-specific visualizations

Galaxy Analyze Data Workflow Shared Data Visualization Help User Using 4.0 MB

Tools

search tools

Get Data

Text Manipulation

Convert Formats

Filter and Sort

Join, Subtract and Group

Prediction

Extract Features

Statistics

Graph/Display Data

Formation

Workflows

- All workflows

Protein Viewer on 'Get PDB' (entity "VIRAL PROTEIN/IMMUNE SYSTEM")

Molecule has 1 models, and following chains: G, B, L, H, D, E.

Information —

Style

- Cartoon
- Balls & Sticks
- Lines
- Line Trace
- Smooth Line Trace
- Trace

Color by

- Uniform
- Succession
- Occupancy
- Temperature factor

Information

Selected [G/GLN287/CA](#)

History

search datasets

Unnamed history

1 shown

2.0 MB

1: Get PDB

~26,000 lines

format: **pdb**, database: ?

HEADER VIRAL PROTEIN/IMMUNE SYSTEM

TITLE CRYSTAL STRUCTURE OF THE HIV

TITLE 2 COMPRISING ATOMIC-LEVEL DEF

TITLE 3 COMPLEX WITH HUMAN ANTIBODI

COMPND MOL_ID: 1;

COMPND 2 MOLECULE: ENVELOPE GLYCOPRO

Molecule sequence

AENLWVTYYYGVPVWIKDAETTLFCASDAKAYETEKHNWATHACVPTDPNPQEIHLNVTVEEFNMWKNMVEQMHTDIISLWDQSLKPCVKLTPLCVTLQCTNV

Note

You can rotate the model by dragging with the left mouse button and zoom by scrolling. You can display the residue number by left-clicking on the molecule.