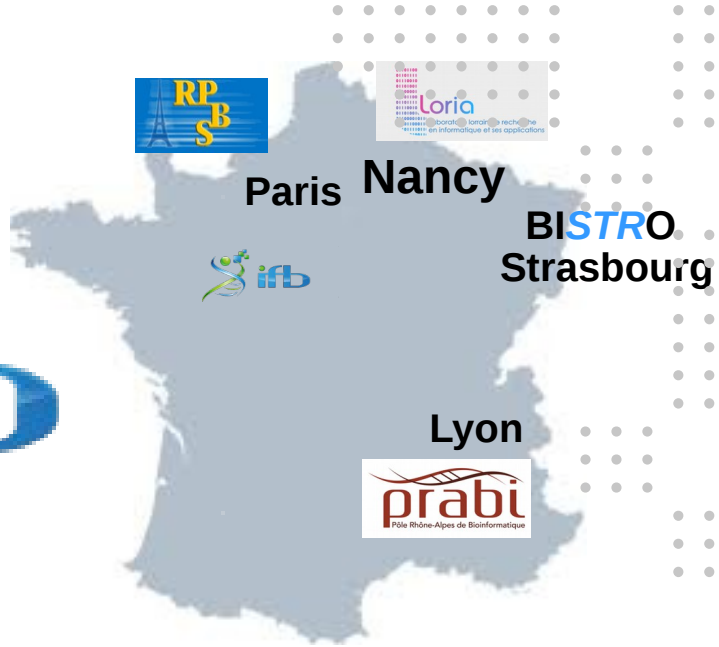


Interfacing PV in Galaxy

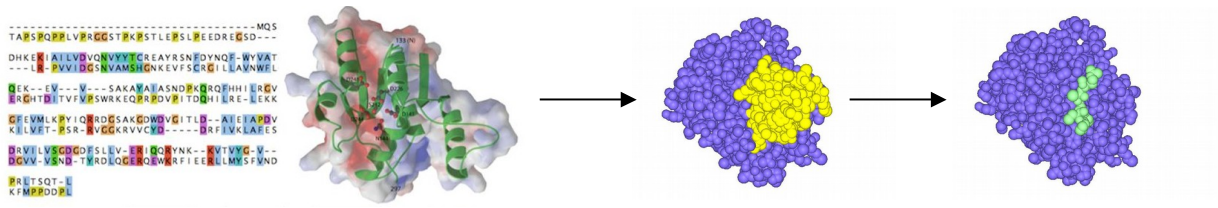
RPBS, U Paris Diderot, 28-Jun-2017

Antoine Chemardin (engineer), Inria Nancy / LORIA

Dave Ritchie (coordinator), Inria Nancy / LORIA

Introduction

3D Visualisation



- 3D visualisation is very important for molecular modeling
- Many high quality desktop tools already exist – e.g. VMD, PyMol, ...
- These all use OpenGL to exploit graphics hardware (nVidia, Radeon, Intel, ...)

3D Web Interfaces ?

- Until recently, the only software available was Jmol / JSmol (non-OpenGL)
  - Web interfaces for 3D graphics were very low quality & slow
- New HTML-5 offers access to hardware through “WebGL” (PV, NGL, webmol, ...)

PV + Galaxy

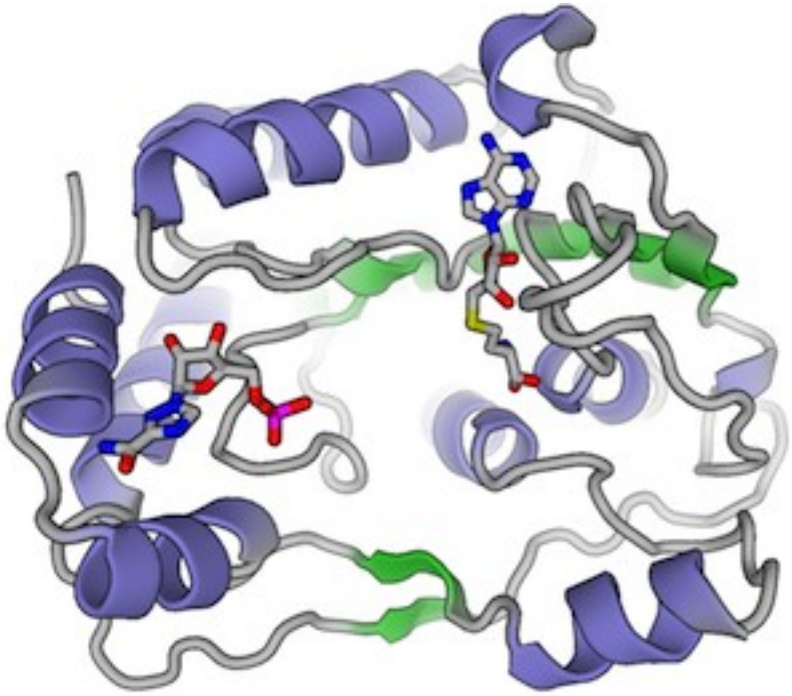
- We chose PV as the “best” WebGL-based viewer for 3D protein structures
- We chose Galaxy as the “de facto standard” bioinformatics work-bench

PV – JavaScript Protein Viewer

https://biasmv.github.io/pv/

About PV

- Developed by Marco Biasini (SwissProt)
- Open source JavaScript + WebGL
- Used in PDB and SwissProt web sites

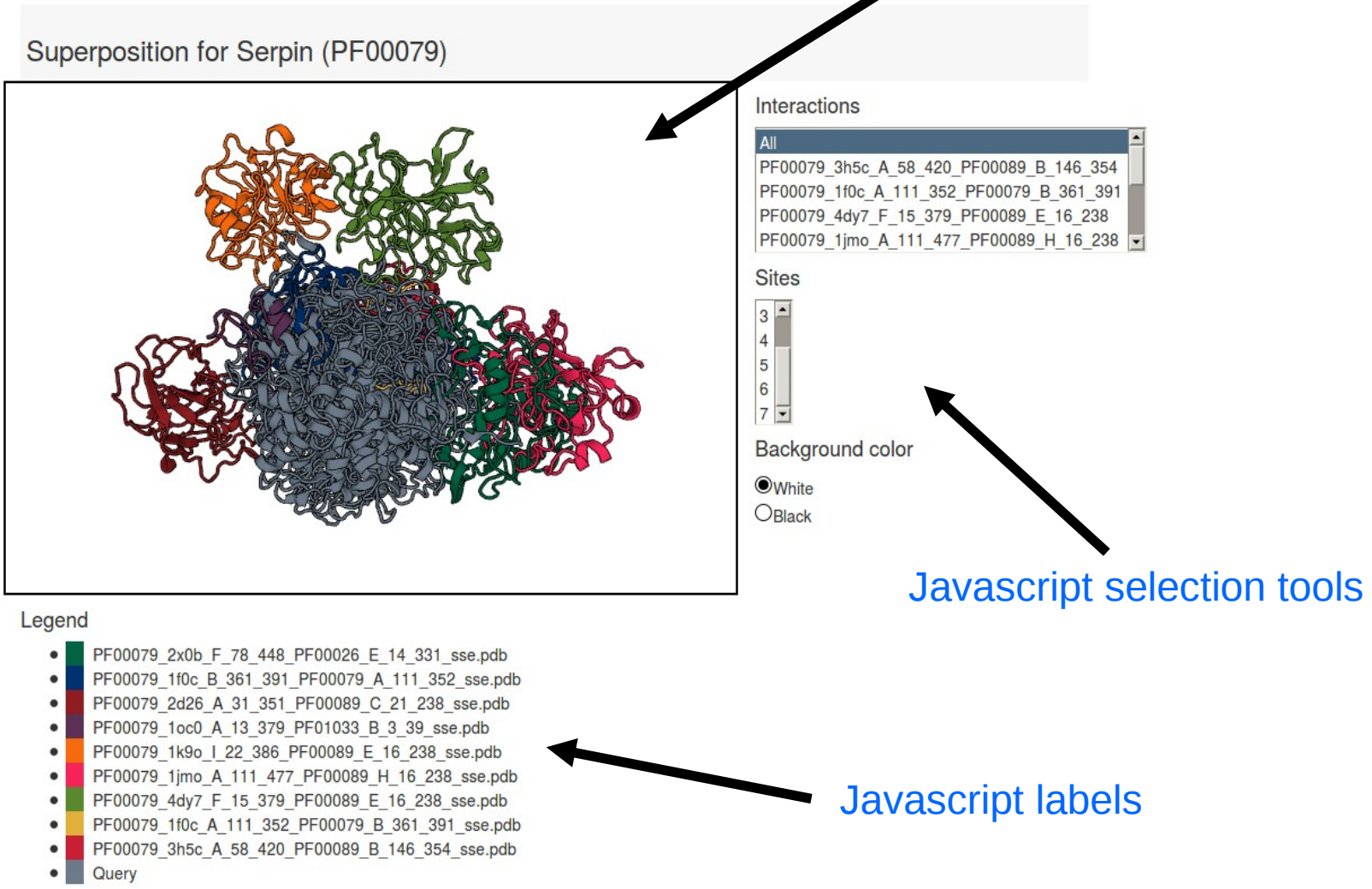


Features

- Good JavaScript API
- Good documentation and example scripts
- Handles multiple proteins simultaneously
- Supports many ways to render proteins:
  - Ribbon cartoons
  - Ball-and-stick
  - … etc.

PV in HTML+JS Web Page (KBdock)

https://kbdock2.loria.fr (no login needed)



PV Programming – HTML Side

In the HTML code “div”s identify the JS parts:

```
<html>
<body>
<div class=”row”>
  <div class=”col-md-8” id=”viewerwrap”>
    <div id=”viewer”></div>
  </div>
  <div class=”col-md-4”>  <select id=”interactions”>
                            <option value=”structure-1.pdb” </option>
                            <option value=”structure-2.pdb” </option>
                            ...
                          </select>
  </div>
</div>
...
</body>
<script src=”../js/pv-bio.min.js”></script>
<script src=”../js/pvviewer.js”></script>
<script> $(document).ready(function() {}) </script>
</html>
```

PV Programming – JavaScript side

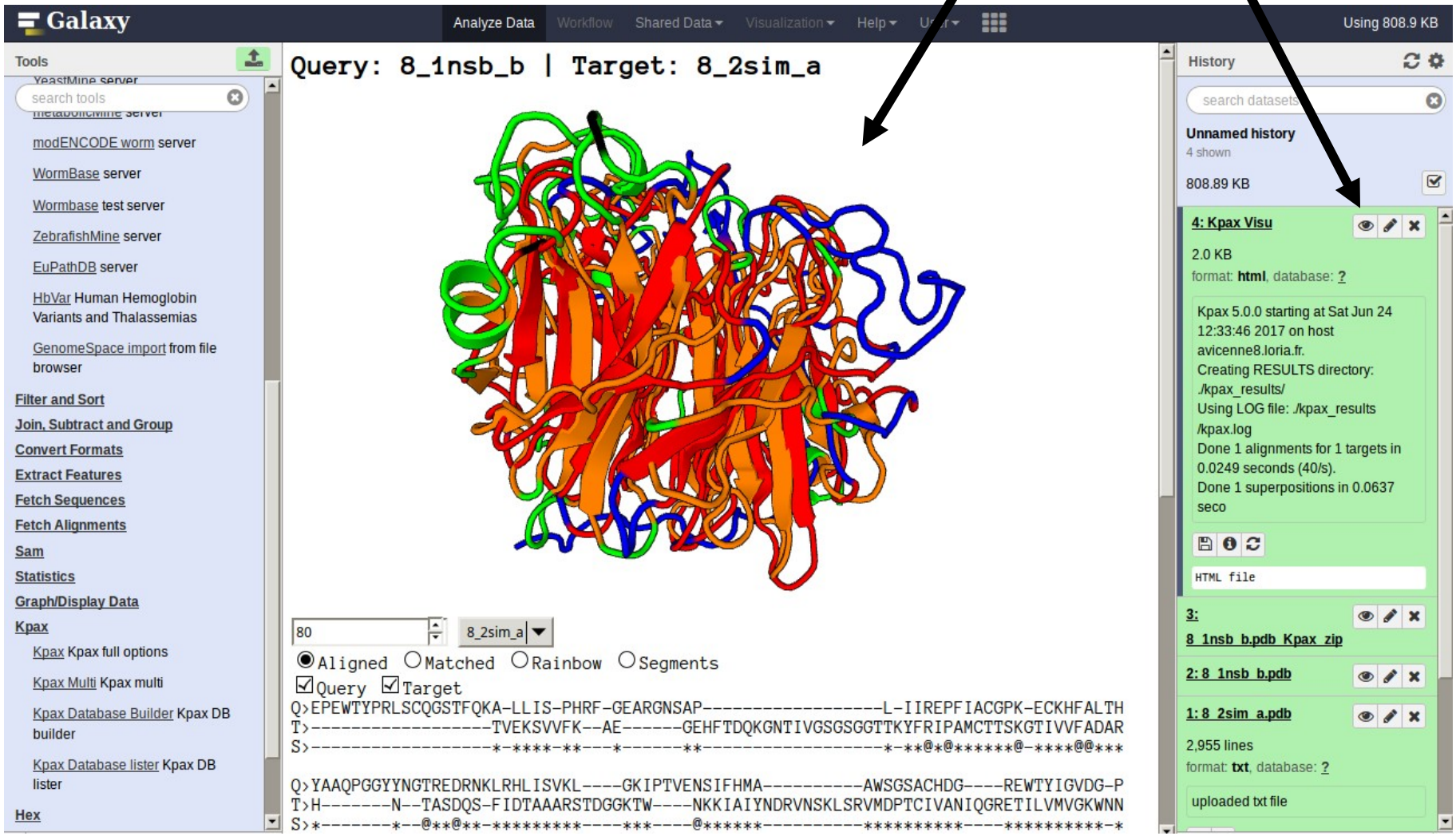
In the JavaScript code (our file ../js/pvviewer.js):

```
// define viewer rendering options
var options = {width: 'auto', height: 435, antialias: true, quality: 'medium',
                    fog: 'false', forceManualAntialiasing: false, background: 'white' };

// insert PV in the div named 'viewer'
var viewer = pv.Viewer(document.getElementById('viewer'), options);

$(function() {          // launch the viewer

    document.getElementById('viewer').addEventListener('mousemove',
                                                        function(event) {
    // more stuff here for handling mouse events …
    });

    pv.io.fetchPdb(prot, function(structure) {
    // more stuff here for loading and processing a PDB file...
    });
});
```

PV Inside Galaxy (Kpax Superposition)

https://galaxy.loria.fr (login required)

HTML results window
(click on “eye” to open)



PV Programming in Galaxy

In Galaxy tool definition, set output format = “html”

Example: /galaxy/tools/kpax/kpax.xml

```
<tool id=”kpax” name=”Kpax” version=”1”>
  <input>
  // define command input parameters and options here (not shown)
  </input>
  <outputs>
     <data format="zip" name="output_zip"
                        label="${query_structure.name}_${tool.name}_zip">
     </data>
     <data format="html" name="html_file" label="Kpax Visu">
     </data>
  </outputs>
  <command>
  // run script to execute Kpax (also generates html results page)
  /kpax/kpaxwrapper.sh $html_file -nosubdirs
                                  -sort="$sort_by" -top="$advanced.top"
                                  -show="$advanced.show" ...
  </command>
</tool>
```

PV – On-Line Resources

PV Base Distribution

https://biasmv.github.io/pv

PV Documentaion & Examples

http://pv.readthedocs.io

PV + Galaxy from Inria Nancy

https://gitlab.loria.fr/capsid/ (request password)

Inria/LORIA Galaxy instance with PV + Kpax + Hex

https://galaxy.loria.fr/ (login required)

PV – Conclusions

Advantages

- Relatively easy to program in JavaScript
- Good documentation & examples
- Can load arbitrary number of protein structures
- Easy to customise (picking, coloring, mouse interaction, ...)
- Original developer responds to questions + bug reports

Disadvantages / Limitations

- Requires JavaScript (which is no longer acceptable for NAR web server)
- Has many advanced features – colour by residue, measure distances, ...
- The API seems to encourage nasty (recursive-style) function invocation
- Needs secondary structure definitions in PDB files
- Does not yet support orthographic viewport...

Thank You !

Questions ?