BIOINFORMATICS APPLICATIONS NOTE Vol. 30 no. 23 2014, pages 3408–3409
doi:10.1093/bioinformatics/btu567

Sequence analysis Advance Access publication August 21, 2014

# Visualization of protein sequence features using JavaScript and SVG with pViz.js

Kiran Mukhyala† and Alexandre Masselot*,†

Department of Bioinformatics and Computational Biology, Genentech Inc., South San Francisco, CA 94080, USA

Associate Editor: John Hancock

## ABSTRACT

**Summary:** pViz.js is a visualization library for displaying protein sequence features in a Web browser. By simply providing a sequence and the locations of its features, this lightweight, yet versatile, JavaScript library renders an interactive view of the protein features. Interactive exploration of protein sequence features over the Web is a common need in Bioinformatics. Although many Web sites have developed viewers to display these features, their implementations are usually focused on data from a specific source or use case. Some of these viewers can be adapted to fit other use cases but are not designed to be reusable. pViz makes it easy to display features as boxes aligned to a protein sequence with zooming functionality but also includes predefined renderings for secondary structure and post-translational modifications. The library is designed to further customize this view. We demonstrate such applications of pViz using two examples: a proteomic data visualization tool with an embedded viewer for displaying features on protein structure, and a tool to visualize the results of the variant_effect_predictor tool from Ensembl.

**Availability and implementation:** pViz.js is a JavaScript library, available on github at https://github.com/Genentech/pviz. This site includes examples and functional applications, installation instructions and usage documentation. A Readme file, which explains how to use pViz with examples, is available as Supplementary Material A.

**Contact:** masselot.alexandre@gene.com

**Supplementary information:** Supplementary data are available at *Bioinformatics* online.

Received on June 3, 2013; revised on August 4, 2014; accepted on August 17, 2014

## 1 INTRODUCTION

Visualization of protein features such as protein domains, post-translational modifications or sequence polymorphisms is critical for interpretation and analysis of proteomic data. Interactive visualization of multiple sources of protein features allows for integrated data analysis and hypothesis generation.

A number of tools have previously been developed that enable visualization of protein features. Some Web sites include custom views that are tightly integrated with the Web site [ex. nextProt (Lane *et al*., 2012), Interpro (Hunter *et al*., 2011), Pfam (Punta *et al*., 2012), Protein Data Bank (PDB) (Berman *et al*., 2000), etc. among many others]. Distributed Annotation System (DAS Dowell *et al*., 2001) client tools like the DASProteinFeatureViewer component of BioJS (Gomez *et al*., 2013) and Jalview (Waterhouse *et al*., 2009) offer limited flexibility to customize the view or data sources, as data needs to be served by a DAS server or uploaded by the user. Most others tools, like Protter (Omasits *et al*., 2013), are Java tools that have dependency on a particular version of Java, require servers for hosting the application and are not designed to be customized or incorporated into a third-party Web site. Tools for customizing the data sources, user interactions or styling of protein features are limited for developers of Web-based visualizations.

pViz.js (pViz) makes it easy to display protein sequence features on a Web page. Knowledge of HTML and basic JavaScript is all that is required to use the library. Basic usage involves including the library within a script tag in HTML, and writing a few lines of code. Given a sequence and a list of positioned features, the library displays the sequence and the features aligned to it. Features are laid out, so as not to overlap each other, allowing zoom, interaction and customization. Developing pViz as a core library allows it to be used as a component of other third-party libraries or Web pages allowing for application specific customization.

## 2 METHODS

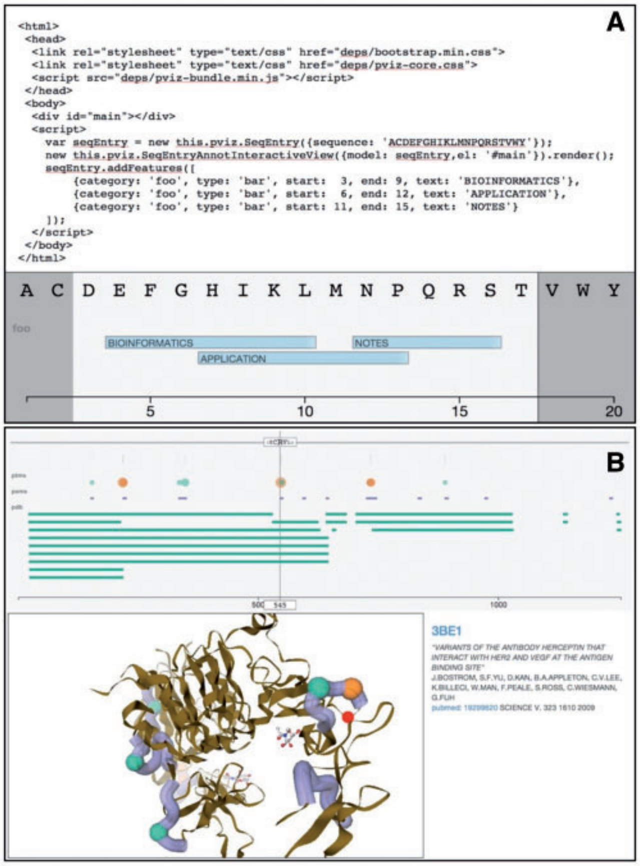
pViz requires writing JavaScript code to create a protein features view. JavaScript is a popular language for developing interactive Web applications and is widely used in bioinformatics. This programmatic approach allows for flexibility in customizing the data sources, display and interactions. pViz is built on other commodity libraries, including backbone.js (http://backbonejs.org), d3.js (http://d3js.org), bootstrap (http://getbootstrap.com) and jQuery (http://jquery.com), which are bundled in the pViz distribution.

A view in pViz is built by creating a *SeqEntry* object, containing a sequence and a list of features (see fig. 1A for a working example). Sequence features have start and end positions and can have several optional attributes: (i) type, which is used for setting display and interaction behavior (e.g. ‘Phosphorylation’, ‘Helix’, ‘PTM’ or ‘SNP’), (ii) category, which groups features and displays them on a separate track, (e.g. ‘secondary structure’ or ‘domain’ and (iii) group, which allows features in the same category separated by their group, (e.g. ‘experiment’, ‘predicted’). Additional attributes can be attached to a feature, such as text to be displayed or a quantification measure, which can be used to customize the display. Sequences and features typically come from many different sources. pViz supports FASTA, DAS and Proteomics Standards Initiative (PSI), PSI Extended Fasta Format.

pViz also supports display of multiple proteins. This can be useful when a simple summary of all features in a set of proteins is desired before presenting features on a single protein. It is this ability of the

*To whom correspondence should be addressed.

†The authors wish it be known that, in their opinion, the first two authors should be regarded as Joint First Authors.



```
<html>
 <head>
  <link rel="stylesheet" type="text/css" href="deps/bootstrap.min.css">
  <link rel="stylesheet" type="text/css" href="deps/pviz-core.css">
  <script src="deps/pviz-bundle.min.js"></script>
 </head>
 <body>
  <div id="main"></div>
  <script>
    var seqEntry = new this.pviz.SeqEntry({sequence: 'ACDEFGHIKLMNPQRSTVWY'});
    new this.pviz.SeqEntryAnnotInteractiveView({model: seqEntry,el: '#main'}).render();
    seqEntry.addFeatures([
        {category: 'foo', type: 'bar', start:  3, end: 9, text: 'BIOINFORMATICS'},
        {category: 'foo', type: 'bar', start:  6, end: 12, text: 'APPLICATION'},
        {category: 'foo', type: 'bar', start: 11, end: 15, text: 'NOTES'}
    ]);
  </script>
 </body>
</html>
```

3BE1

"VARIANTS OF THE ANTIBODY HERCEPTIN THAT INTERACT WITH HER2 AND VEGF AT THE ANTIGEN BINDING SITE"

J.BOSTROM, S.F.YU, D.KAN, B.A.APPLETON, C.V.LEE, K.BILLECI, W.MAN, F.PEALE, S.ROSS, C.WIESMANN, G.FUH

pubmed: 19299620 SCIENCE V. 323 1610 2009

**Fig. 1.** (**A**) A complete example of using pViz, to generate the graphic shown. (**B**) A screenshot of an application of pViz showing features on protein sequence and structure. See Supplementary Material B for detailed descriptions

library to let developers adapt it to their needs that makes pViz a powerful tool. Both the display (size, shape or color) and interaction (the action performed on a mouse event) for each feature type can be customized based on the needs. Display can be defined via the cascading style sheets rules or explicit construction of graphical widgets in Scalable Vector Graphics (SVG). Although customizing SVG is more complex, it is supported by the d3.js library, which allows constructing virtually any type of graphical widget.

Interactivity is a key feature of pViz. The default implementation allows zooming on the sequence and one can define behaviors triggered by a mouse click or hover position on the sequence or features. Standalone examples of these functionalities are included in the distribution. To demonstrate these capabilities, we describe two sample applications included in the pViz distribution.

### 2.1 Viewer for Ensembl's Variant Effect Predictor

We developed a viewer for Ensembl's Variant Effect Predictor (VEP; McLaren *et al*., 2010) using pViz, available at http://research-pub.gene.com/pviz/app/vep. The viewer displays the protein coding variants as features on the proteins listed along with any variant annotation in the VEP output. The consequences calculated by the VEP script are used to categorize the variants annotations available in VEP, including protein domains, Sorting Intolerant from Tolerant (SIFT) and Polyphen scores are used to color the variants. The gene name available in the output is used for searching the proteins.

### 2.2 Viewer for Proteomics data on Protein Structures

We also developed a viewer for displaying proteomic data on primary sequence and tertiary structures by extending the GLmol library (webglmol.sourceforge.jp). The mapping of a protein sequence to a PDB structure is parsed from a Uniprot (Magrane and UniProt Consortium, 2011) entry and is displayed as a feature. The customization capabilities of pViz were extended to display the PDB structure when the features were clicked and also to display overlapping features on the protein structures. The mouse-over event was also customized to show the cursor position on the sequence and the structure simultaneously (Fig. 1B). This tool is available at http://research-pub.gene.com/pviz/app/proteomics-3d

## 3 CONCLUSION

There has been a lack of tools for developing interactive protein feature visualizations that are easy to embed in a Web page. pViz is an open-source library that addresses this problem and makes it easy to integrate within larger applications. By using JavaScript, it renders visualization on the client side, thereby avoiding the need for a separate server or a dependency on Java. The library provides basic interactivity like zoom and mouse-over information out of the box with additional hooks for interactions for the developer to implement. Multiple working examples of these, ranging from simple features to highly customized visualizations are available on GitHub.

## ACKNOWLEDGEMENTS

The authors thank Melanie Huntley, Donald Kirkpatrick, Tobias Maile, William Young and Gerard Manning for their feedback during the development of pViz and helpful comments on the manuscript.

*Conflict of interest*: none declared.

## REFERENCES

Berman,H.M. *et al*. (2000) The protein data bank. *Nucleic Acid Res*., **28**, 235–242.

Dowell,R.D. *et al*. (2001) The distributed annotation system. *BMC Bioinformatics*, **2**, 7.

Gomez,J. *et al*. (2013) BioJS: an open source JavaScript framework for biological data visualization. *Bioinformatics*, **29**, 1103–1104.

Hunter,S. *et al*. (2011) InterPro in 2011: new developments in the family and domain prediction database. *Nuclei Acid Res*., **33**, W116–W120.

Lane,L. *et al*. (2012) neXtProt: a knowledge platform for human proteins. *Nucleic Acids Res*., **40**, D76–D83.

Magrane,M. and UniProt Consortium (2011) Uniprot Knowledgebase: a hub of integrated protein data. *Database* bar009.

McLaren,W. *et al*. (2010) Deriving the consequences of genomic variants with the Ensembl API and SNP Effect Predictor. *Bioinformatics*, **26**, 2069–2070.

Omasits,U. *et al*. (2013) Protter: interactive protein feature visualization and integration with experimental data. *Bioinformatics*, **30**, 884–886.

Punta,M. *et al*. (2012) The Pfam protein families database. *Nucleic Acid Res*., **40**, D290–D301.

Waterhouse,A.M. *et al*. (2009) Jalview Version 2-a multiple sequence alignment editor and analysis workbench. *Bioinformatics*, **25**, 1189–1191.