Batch mode generation of residue-based diagrams of proteins

Fabien Campagne¹, *, Emmanuel Bettler², Gert Vriend² and Harel Weinstein¹

¹Institute for Computational Biomedicine and Department of Physiology and Biophysics, Mount Sinai School of Medicine, Box 1218, New York, NY 10029, USA,
²Centre for Molecular and Biomolecular Informatics, University of Nijmegen, Toernooiveld 1, P.O. Box 9010, 6500 GL Nijmegen, The Netherlands

Received on December 31, 2002; revised on March 6, 2003; accepted on April 2, 2003

ABSTRACT
Summary: Residue-based diagrams of proteins are graphical representations that can be used in protein information systems. These diagrams make it possible to visually integrate different types of biological information. The approach has been used successfully for membrane proteins. We developed the Residue-based diagram generator to (i) make it possible to generate residue-based diagrams of proteins in a batch mode that is compatible with the needs of information system curators, (ii) allow the generation of residue-based diagrams for families of soluble proteins or domains.

Availability: Licensed. Royalty free licenses are granted to non-profit institutions for educational and research purposes. http://icb.mssm.edu/crt/RbDg/index.xml
Contact: Fabien.Campagne@mssm.edu

INTRODUCTION
Residue-based diagrams of proteins are a special kind of protein visualization. In such diagrams, residues of the protein are drawn according to graphical conventions that encode the secondary structure elements of the proteins. Since residues are shown individually on the diagram, they can appear with associated properties. This is especially important for membrane proteins for which 3D structural information is still scarce, but can be useful as well for other protein families. As an illustration, Figure 1 presents a residue-based diagram of the human estrogen receptor beta. Several programs have been developed to automate the rendering of diagrams for membrane proteins, or assist in their drawing. These programs include Viseur (Campagne, 1995; Campagne and Jestin, 1999, http://www.lctn.u-nancy.fr/viseur/), VHMP (Lin and Hwang, 1998), the Residue-based diagram editor (RbDe) (Campagne and Weinstein, 1999; Konvicka et al., 2000) and TExtopo (Beitz, 2000).

The Viseur program and RbDe are unique among these programs in their ability to generate diagrams embedded in an HTML page. Residues shown on the diagram can be hyperlinked to specific documents on the web. This feature made it possible to use diagrams generated by the Viseur program for data integration in the GPCRDB (Horn et al., 1998). The GPCRDB uses hyperlinked residue-based diagrams to integrate sequence data with mutant information or with results from correlated mutation analysis. The GPCRDB currently describes more than 4000 GPCR sequences, emphasizing the need for a completely automated process to generate batches of residue-based diagrams. Such batch updates must be performed for each database release since a diagram must be regenerated whenever there is a change in the data from which it is calculated (e.g. new mutations enter the database, or new sequences enter the alignment used to calculate correlated mutations).

*To whom correspondence should be addressed.
RbDe, which has proven useful in the interactive construction of residue-based diagrams, does not support the batch generation needs of information systems such as GPCRDB. The availability of RbDe has stimulated interest in ways of creating residue-based diagrams to support the data integration needs of information systems for other (non-membrane) proteins, such as nuclear receptors (Horn et al., 2001), or WW domain-containing proteins (Macias et al., 2002; Sudol, M. personal communication). To address these new requirements, we have designed and implemented a program that can be used in batch mode. We called this program the Residue-based Diagram generator (RbDg).

RESULTS
RbDg requires input files that describe the sequence and the secondary structure segmentation of the protein. In addition to these data, and in order to support a variety of protein families, users must describe how the secondary structure elements are to be laid out on the page (that is, their relative positions on the page). The layout is described by a succession of secondary structure element (SSE) types (i.e., helix, beta strand), and spacer vectors for the loops that separate these elements (see Campagne and Weinstein, 1999) for the definitions of SSE and spacers. The algorithm implemented in RbDe optimizes the shape of loops based on several parameters, including the positions at which ellipsis glyphs can be introduced. Ellipsis glyphs replace several contiguous residues with a graphical symbol that indicates that a portion of the sequence is not shown (typically the icon used by RbDe is ‘…’ in a circle). This feature makes it possible to fit the important residues in the diagrams, while hiding others, and is key to defining a layout that can render many proteins of the same family, regardless of variations in the length of the loops among members of the family. Since RbDg leaves the choice of the important residues to the database curators, positions considered to be important can be stored in the information system or calculated from primary data (for instance, positions where mutagenesis experiments have been performed are often considered important).

Other practical features described in the supplementary material (http://icb.mssm.edu/crt/RbDg) make RbDg useful to curators of biological information systems. These include customization of the appearance of the HTML output through XSL templates.

End-users will appreciate that RbDg can produce diagrams in the Scalable Vector Graphics (SVG) format (the GIF image format is also supported). This format can be printed at high-resolution and edited with a growing number of WYSIWYG graphics editors (e.g. Illustrator 10+).

USING RbDg WITH PROTEIN FAMILIES
The following procedure can be used to generate diagrams for every sequence in a protein family, based on a reference sequence for which secondary structure limits are known. Assuming that secondary structure limits are conserved across the family, secondary structure limits can be derived for each protein, and used to construct the ‘<protein>’ section of an RbDg input file. When combined with a diagram layout, and with diagram output options, valid RbDg files can be created and processed.

IMPLEMENTATION
RbDg was implemented in Java and tested on SGI IRIX, Linux, Solaris Intel and SPARC workstations. Generation of one nuclear receptor diagram, as illustrated on Figure ‘Diagram Generation Process’, on the web site takes about 1 s of CPU time on a single processor Linux workstation.

CONCLUSION
Based on its tested useful properties, RbDg will be advantageous for use in information systems (e.g. GPCRDB, etc.) because it provides annotated diagrams that can be further customized by the end-user for specific publication needs. Since SVG diagrams can be fully hyperlinked (on each residue and secondary structure unit), users could also customize their own residue-based diagram for publication on the web.

ACKNOWLEDGEMENTS
This work was supported in part by NIH grants P01 DA124080, P20 HL69733 and by Organon and Unilever.

REFERENCES