Biochemfusion

Rendering info text

Format description

version 1.0

2010-11-12

Table of Contents

Introduction

Biochemfusion rendering info text is a simple way of representing drawing primitives in a single line of text. Biochemfusion's Proteax software toolkit uses rendering info text to graphically render protein sequences and 2D molecule diagrams with a minimum of output device constraints.

A valid alternative approach would be to provide graphical renderings as SVG files instead. However, XML parsing will usually introduce quite a lot of parsing overhead as well as a large increase in application footprint for some systems, e.g. Flash-based renderers.

Also, the size of an SVG file will be much larger than the corresponding rendering info text string. This is a vital issue if you attempt to store or exchange rendering information in situations where there is a hard limit on maximum string size, e.g. spreadsheet cells.

1 General format

Rendering info text consists of colon-separated tokens. The first token will be a fixed header while the rest of the tokens will be two-letter tags that may have associated comma-separated data. The rendering info text is always terminated by a fixed end tag.

```
    :<header>:<tag1>[data]:<tag2>[data]: ... <tagn>[data]:<end tag>
```

2 Protein sequences

Rendering info for protein sequences starts with the header tag `BCF_SEQ_RNDR_INFO_V1` and ends with the end tag `ED` (End of Data).

2.1 BB - bounding box tag

```
BB<cell-size>,<line-spacing>,<residues-per-line>,<line-count>,<residue-count>
```

The bounding box data defines the virtual coordinate space used. Each residue occupies a cell that has a dimension of *cell-size* x *cell-size* coordinate units. The first residue is placed in the cell having its upper left corner at (0, 0) and its lower right corner at (*cell-size*- 1, *cell-size* - 1). Each line of residues is vertically separated by a blank space which is *line-spacing* units high.

The remaining information can then be used to calculate the full bounding box of the sequence graphics. The Biochemfusion rendering tools will make additional room for e.g. placing residue numbers to the left of the sequence.

2.2 CH - chain tag

```
CH<chain-number>,<terminal-flags>,<sequence-data>
```

chain-number is an integer that is 0 if the chain is a non-expressed chain aka. an exo-chain. Otherwise the *chain-number* identifies the chain, chain "A" having number 1, chain "B" number 2, etc... Of the formats supported by Proteax, exo-chains can only occur in UniProt files.

terminal-flags consists of two characters that indicate whether respectively the N-terminal and the C-terminal is modified. '0' means not modified, '1' means modified.

sequence-data contains the one-letter residue codes for the chain. Modified residues are prefixed with a numeric one-digit code. The prefix code is a 2-bit bitmap where a value of 1 means a modified residue, and a value of 2 means a D-form residue; thus a value of 3 means a modified D-form residue.

2.3 SS - disulfide tag

```
SS<from>,<to>,<comma-separated (x,y) coordinate pairs>
```

Indicates the presence of a disulfide bridge going from the *from* residue to the *to* residue (absolute residue numbers). This information is not needed for graphical rendering unless you wish to calculate a drawing path yourself.

The trailing (x, y) coordinate pairs list Proteax's suggested path for drawing the disulfide bridge.

2.4 XL - crosslink tag

```
XL from, to, <comma-separated (x,y) coordinate pairs>
```

Indicates the presence of a cyclization or crosslink going from the *from* residue to the *to* residue (absolute residue numbers). This information is not needed for graphical rendering unless you wish to calculate a custom crosslink drawing path.

The trailing (x, y) coordinate pairs list Proteax's calculated crosslink drawing path in cell units.

2.5 Tag order

Protein sequence rendering info text shall list tags in this order: bounding box, chains, disulfide bridges, crosslinks.

3 2D molecules

Rendering info for 2D molecule diagrams starts with the header tag `BCF_MOL_RNDR_INFO_V2` and ends with the end tag `ED` (End of Data).

The molecule rendering info that starts with `BCF_MOL_RNDR_INFO_V1` was used by Proteax for Spreadsheets 1.1 2010-01. This version was for internal-use only and will not be documented.

(x, y) coordinates are angstrom (or rather: MDL molfile[1]) coordinates multiplied by a scaling factor and rounded to produce integer coordinates. The reason for using integer coordinate values is to avoid parsing problems due to differing regional settings. Coordinates have been transformed so the Y-axis direction matches standard pixel devices (increasing Y-values going down) and translated to fit inside the calculated bounding box.

The coordinate scaling factor will be chosen by the producer of the rendering info text to strike a sensible balance between rendering info text length and coordinate resolution. The current Proteax release uses a fixed value of 100.

3.1 BB - bounding box tag

```
BB<coordinate-scaling-factor>,<x-max>,<y-max>
```

Three integers define the coordinate scaling factor and the maximum (x, y) coordinates of the molecule bounding box. The bounding box always has an origo of (0, 0).

3.2 BD - bond tag

```
BD<x1>,<y1>,<x2>,<y2>,<aa>
```

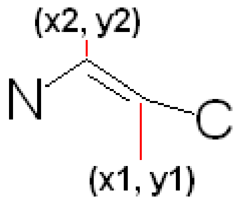
Implementations should draw a bond from (*x1*, *y1*) to (*x2*, *y2*). The *aa* data is a two-letter string that defines bond cardinality and configuration. The possible *aa* values are listed below.

Value	Description
`1.`	Single bond.
`1u`	Up stereo bond.
`1d`	Down stereo bond.
`2s`	Double symmetrical bond.
`2l`	Double bond - left.
`2r`	Double bond - right.

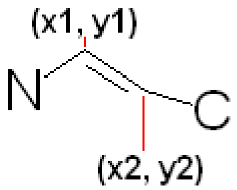
`3.`	Triple bond.

In addition to the above values *aa* can be "`a`" followed by an integer from 1 to 9. This indicates an attachment point with the given number.

The double bond "left" and "right" orientations specify how an asymmetrical double bond should be drawn. The shorter "free-floating" bond will be drawn to the left or right of the single bond as seen from (*x1*, *y1*) towards (*x2*, *y2*). See below for examples of drawing an asymmetrical double-bond having different directions.



"Left"



"Right"

3.3 AT - atom tag

```
AT<x>,<y>,<atom-symbol>[,<extra-info>]
```

Implementations should place an atom symbol label at position (*x*, *y*). It is the implementation's responsibility to place the label centered around (*x*, *y*) - the rendering info text producer will make no assumptions about font sizes etc.

The optional extra information defines special atom display properties, e.g. charges, implicit hydrogens, isotopes, etc.

If the atom is a condensed residue atom, e.g. "Ala", the extra information will be an integer that provides the number of the chain that the residue atom is located in. This may be used by renderers to color residue atoms so chains are visually grouped.

Charges are recognizable as explicitly signed integers, e.g. "+1" or "-2". Positive charges will always be prefixed with a plus-sign.

Implicit hydrogens are represened by "H*on*" where *o* is a single character defining the hydrogen display orientation relative to the atom symbol; **b**elow, **a**bove, **l**eft, **r**ight, and *n* is an integer defining the number of hydrogens.

Representation of future extra information, like isotopes, is currently undefined but implementations should allow/ignore future extra information. Charges and implicit hydrogens will be listed before any future extra information; this means that an implementation that only supports e.g. charges and implicit hydrogens can safely ignore remaining unrecognizable data.

NOTE: At the time of writing Proteax will output residue atom chain numbers and charges only.

3.4 *LB - label tag*

```
LB<x>,<y>,<alignment>,<label>
```

Implementations should place the text *label* at the point (*x*, *y*).

alignment defines the horizontal text alignment relative to (*x*, *y*). Vertical alignment is always middle/center.

Value	Description
`l`	Left aligned - text starts at (x, y).
`r`	Right aligned - text ends at (x, y).
`c`	Center aligned - text centered around (x, y).

label is a text that will have the following characters percent-encoded.[2]

Character	Percent-encoded value
`%        [percent]`	%25
`,        [comma]`	%2C
`:        [colon]`	%3A

Percent-encoded values will always use uppercase letters, never lowercase letters.

The use of percent-encoded label captions instead of quoting simplifies the parsing of rendering info text. Any occurrence of a colon is guaranteed to mark the beginning of a new tag and any occurrence of a comma will always mark the beginning of a new value.

NOTE: Spaces and any other special characters will not be encoded but just listed literally. Newlines and other control or formatting characters should never occur.

3.5 *Tag order*

Molecule rendering info text shall list tags in this order: bounding box, bonds, atoms, labels.

4 References

(1) The MDL molfile format was defined by the company MDL, then Symyx, now Accelrys. The molfile specification is public and available at http://www.mdli.com/downloads/public/ctfile/ctfile.jsp.

(2) Percent-encoding is commonly used for URI data. See http://en.wikipedia.org/wiki/Percent-encoding.