

Optimal production of biological documentation: the JAM format

R.Doelz

Abstract

The current environment for providing documentation for users in molecular biology frequently requires written information for both printed and electronic media. The JAM (Just Another Metafile) Format requires that only a single file of text is written. This is, then, processed by the jam program which converts this text to a document set which is suitable for on-line documentation in the hypertext mark-up language (HTML) of the World Wide Web system. For printed output, the jam program can produce either text in wordprocessor-ready Rich Text Format (RTF), or source code for the LaTeX typesetting system. The latter allows for professional text layout and can generate, amongst other formats, postscript files to be printed on a variety of devices.

Today, documentation in the biocomputing disciplines must be available in various formats. Users who have never used computers for biocomputing purposes before will not easily adopt all-electronic information and require professionally printed material. Colleagues at remote sites who wish to print the text locally in chapters, or send it around in electronic Mail messages, require a reasonable format to communicate structured text which can be further processed on-site. Electronic hypertext systems, on the other hand, have become increasingly popular in recent years and have the tremendous advantage that the cross-references are much more intuitive and easy to follow with sophisticated browsers such as Mosaic (a program that interprets and displays hypertext as formatted 'pages', and allows to follow links with mouse clicks).

For documentation that has been created in the biology user support sector, the use of translators provided a preliminary way of switching between different media. Various elements of text structuring and cross references are similar between the printed and electronic formats, and allow translators to achieve a one-to-one imaging of the text formats. However, the detailed wording and presentation to the user are different and depend on the context of the document. Some information is lost during

the translation, and the reader might have difficulty to understand details.

The required formats include the plain text as used in word processors on personal computers, text prepared for the LaTeX typesetting system, and 'hypertext mark-up language', HTML. The following example illustrates the desired appearance, which cannot be achieved by translators:

ASCII TEXT

This program has been discussed earlier.

LaTeX source (usually not visible to the user)

This program (see also, `\ref{seqed}`) has been discussed earlier.

LaTeX appearance:

This program (see also, 3.1.5) has been discussed earlier.

HTML source (usually not visible to the user)

This ` program ` has been discussed earlier.

HTML appearance

This *program* has been discussed earlier.

To avoid the obvious shortcomings of translators, which were unable to create the explanatory wrappers for crosslinks as depicted above, we have created the JAM format in order to produce different formats from one single source. This source can be edited and handled independently and does not require that either of the tools for formatting (Word processor, LaTeX or WWW browser) is available during editing.

The Just Another Metafile (JAM) format

This JAM format is suited to be translated into either LaTeX, hypertext markup language (HTML), or Rich Text Format as commonly accepted by word processors. An alphabetical index as well as a table of contents is provided within each different format.

All resulting presentation takes full advantage of the specific capabilities of the word processing system used.

LaTeX processing

The *jam* program produces a LaTeX source file which can be processed with the usual LaTeX procedures. LaTeX

processing requires an additional stylesheet which comes with JAM in order to be processed, and the *makeindex* utility to generate the correct index (*makeindex* is part of each LaTeX installation).

On-line version in HTML

The *jam* program produces a set of files which can be viewed after proper installation with any WWW client. Both LYNX and Mosaic type of viewers are supported, i.e. there is no need for sophisticated FORMS support. *Jam* allows specific hypertext links for referencing electronic sources.

Electronic/printed version in RTF

The RTF version generated from JAM Format has a running head for nice printing in word processors. Hypertext links outside of the document are not resolved but hidden text elements are available to create both index and table of contents if this feature is available within the word processor.

Other formats

The *jam* program implementation is clearly structured and easily allows the adaptation of other formats such as plain text (ASCII, not covered in the distributed version).

JAM processor availability

The *jam* program has been implemented on the OSF/1 flavour of the UNIX operating system and was successfully compiled on IRIX, ULTRIX, AIX, Solaris, AXP/VMS, VAX/VMS, Microsoft Windows 3.11 and Macintosh 7.1 operating systems. The JAM Format has been developed at the Biocomputing Facility in Basel over the years and was put into the public domain in 1994. The code is available from the author (doelz@comp.bioz.unibas.ch) on request, or via anonymous ftp from bioftp.unibas.ch in the programs/bioftp-sw/jam directory. Binaries as well as source code are available. In order to recompile the code with the graphical user interface, the VIBRANT toolkit (Kahns *et al.*, NCBI) must be installed.

Application of the JAM Format

The 'Biocomputing Survival Guide' is a cookbook which we have prepared at our site for the computer novice. It allows users unfamiliar with molecular biology computing to work in our environment effectively and with ease. We have updated the guide as version 2.0 and rewritten it in JAM format. The Guide can be translated into either a VMS or a UNIX version, and allows for site-specific

details within more than ten include files. The overall material to be changed to adapt the guide to a given site is less than five pages and can be modified with great ease.

We have further created a 'teaching application' using the JAM format. Courses which we offer at the Biozentrum include 'question sheets' and 'answers'. The answers explain how to approach the problems mentioned in the 'question sheets' using the 'Biocomputing Survival Guide' as a reference. The 'Biocomputing Survival Guide', the 'question sheets', the 'answers' and all reference data required to cross-check the obtained results are available via anonymous ftp on bioftp.unibas.ch in the archive_data/survival directory tree in either RTF or LaTeX format. The HTML versions are also on-line as hypertext version, on the URL (universal resource locator) <http://beta.embnet.unibas.ch/basel/course.html> you will find a page to guide you to the various parts of the teaching and documentation section.

Customization

In order to utilize documentation containing site-specific information, extensive editing is usually required. The JAM text source, however, may contain several configurable files which hold site-specific information such as contact addresses for the local computing centre, which editor to use, and similar localized elements. The local information, as well as the residual text, therefore allows the adaptation of individual elements in a single fashion, i.e. both UNIX- and VMS type of environments may be described in the same text file. In order to adapt a document for different environment than the one it was originally developed for, it is sufficient to modify only site-specific files which are included into the document by the *jam* program. Both 'Biocomputing Survival Guide' and teaching material, therefore, may be easily adapted to fit individual needs.

JAM Format specification

The format has several classes of statements, which are processed in the following order:

Comments

The percent (%) character in column 1 of a line indicates a comment.

Conditional statements

The processing directives are extremely simple and similar to the C-pre-processor and expanded before the formatting statements. The use of `#ifdef`, `#else`, and `#endif` statements allows several configurations in the same

format (i.e. VMS and UNIX) to be written in the same JAM source code. Another conditional statement is the `#include` directive that permits site-specific text to be written in separate files, which allows a readily available general version to be 'adapted' to specific sites easily.

Formatting statements

It is required to display four different types of non-flow text. These are

- verbatim (e.g. all is printed as written)
- itemized (such as this text)
- special (e.g. different appearance but not verbatim, used for 'examples')
- explicit links (most useful for the HTML version)

Each of these formatting statements inserts a break in the current text flow and a new line starts after each of these formatting statements. For example, verbatim is printed as written and all formatting of the original text will be retained. Special sections are for examples and similarly emphasised text. Itemized sections do not honour formatting, but rather build a list-like appearance. Explicit Link sections display a hypertext link in the HTML version, whereas in the text version the link is only described in writing.

Structuring statements

Structuring statements are processed after the conditional and formatting statements and are automatically numbered. The text structure permits three levels. Chapters, Sections, and Headlines are implemented.

Referencing statements

The purpose of referencing statements is to allow indexing and transparent reference insertion. Index entries, pointers to anchors and anchors are allowed.

Detailed syntax descriptions, hints, recommendations and examples how to implement various elements of specific text are included in the documentation of the program distribution.

Received January 20, 1995, accepted February 6, 1995