# SWISSKNIFE - "Lazy Parsing" of SWISS-PROT Entries

Henning Hermjakob, Wolfgang Fleischmann, Rolf Apweiler
EMBL Outstation Hinxton
The European Bioinformatics Institute (EBI)
Wellcome Trust Genome Campus
Hinxton, Cambridgeshire CB10 1 SD, U.K.
hhe@ebi.ac.uk

SPTR, the combination of SWISS-PROT and TrEMBL (Bairoch and Apweiler, 1999) forms a comprehensive protein database which combines the high quality of annotation in SWISS-PROT with the completeness of TrEMBL. Use of SPTR for bioinformatics often involves extracting partial information from a huge number of entries in SWISS-PROT text format. While SRS (Etzold *et al.*, 1996) provides a powerful query interface for text-based biological databases, in is not very suitable for extracting partial information or for modifying entries in SWISS-PROT format. Scripting languages like Perl (Wall and Schwartz, 1991) allow fast and easy parsing of text-based databases. However, ad-hoc developed parsers tend to neglect details of the formatting of the parsed database, and it is often difficult to maintain a multitude of specialised parsing statements for a given database.

To overcome these difficulties, we have developed SWISSKNIFE, a set of Perl modules, which provides a fast and reliable object-oriented interface to parsing and modifying files in SWISS-PROT format. If `$entry` is an entry object, an array of the keywords is returned by `$entry→KWs→elements`, and a keyword is added by `$entry→KWs→add('Keyword')`.

A parser which completely parses complex SWISS-PROT entries tends to be relatively slow, and it is computationally expensive to completely parse an entry just to get the list of accession numbers. Therefore we have developed the concept of "lazy parsing" for SWISSKNIFE. When an entry is read from the flat file, it is only stored as an unstructured text. If a particular subobject is requested, e.g. `$entry→ACs`, the text is parsed, the AC object is created and stored as a subobject of `$entry`. All subsequent accesses are then redirected to the subobject instead of to the textual representation.

The concept of "lazy parsing" has allowed us to implement a complete interface to entries in SWISS-PROT format while retaining a high parsing speed if only a few items of the entry are requested. A full parse of 243 SWISS-PROT sample entries requires 2.52 seconds on a Pentium 266, while extracting the accession numbers requires only 0.40 seconds.

### Availability

The SWISSKNIFE modules are available at `ftp://ftp.ebi.ac.uk/pub/software/swissprot/`

### Acknowledgements

This work was supported in part by grant number B104 - CT97 - 2099 of the European Commission. The SWISSKNIFE interface has been modelled on the example of the PrEMBL modules (Pocock *et al.*, 1998).

## References

Bairoch, A. and Apweiler, R. (1999). The SWISS-PROT protein sequence data bank and its supplement TrEMBL in 1999. *Nucl. Acids Res.* **27**, 49–54.

Etzold, T., Ulyanov, A., and Argos, P. (1996). SRS: Information retrieval system for molecular biology data banks. *Methods in Enzymology,* **266**, 114–128.

Pocock, M., Hubbard, T., and Birney, E. (1998). SPEM: a parser for EMBL style flat file database entries. *Bioinformatics,* **14** (9), 823–824.

Wall, L. and Schwartz, R. L. (1991). *Programming Perl.* O'Reilly & Associates, Inc.