



Department of Physiology & Biophysics,
Mount Sinai School of Medicine,
One Gustave L. Levy Place,
New York, NY 10029 USA
Email: campagne@inka.mssmedu

This work has been supported by the
Institute for Computational Biomedicine
<http://icb.mssmedu>

Sequence representation: a comparison of prototypes

Fabien Campagne

Introduction

Bioinformatics analysis of sequence data generally involves the use of several methods on a set of data. Given the increase of the quantity of data and the need to develop new methods quickly to help organize and make sense of the growing flow, there is increasing interest in the development of methods enabling the interoperability of bioinformatics tools [1-4].

The most popular solution, proposed in the bioinformatics community for solving this problem, has been the development of canonical representations of the data. A canonical representation is a unique way of representing the data that has to be shared by all tools that can operate on the data. One form of canonical representation is the file format. All the tools that share a specific file format can potentially interoperate.

Object-oriented data representations strongly bind the data and the methods used to manipulate it, and when used appropriately, help reduce the code duplication that seems to be associated with approaches which rely on files.

Several object-oriented representations of sequence data have emerged recently (see Framework presentation below).

Here, I compare the representations used by four frameworks to find common design patterns and major differences, and attempt to relate the design choices to the amount of interoperability achieved. The frameworks themselves are then compared according to a clearly defined benchmark.

Framework prototypes

The following object-oriented frameworks will be compared here. All of them are at the prototype stage.

BSA [5]
Biomolecular Sequence Analysis (BSA) [?] is the response of a consortium to the OMG Request For Proposal for Life Science Research. Two implementations of the BSA specification are now being developed. This framework is based on the CORBA infrastructure.

BioPerl [6]
An open source project which offers a bioinformatics framework for Perl programmers.

BioJava [7]
An open source project which offers a bioinformatics framework for Java programmers.

Crover
A sequence representation project that was initiated in the spring of 1998 and was initially supported by the Centre Charles Hermites, Nancy, France [8]. It builds on the experience obtained with the Viseur project [9-10]. The development of the framework is being pursued at Mount Sinai School of Medicine. The crover representation was designed to store sequences and annotations such as secondary structure, domains, disulfide-bridges, URL attached to specific residues, etc. to be used for the construction of residue-based visualizations. In addition, this representation has been used in some of our recent work [11-13] as well as in sequence analysis and mining projects, so that we can now step back and evaluate the effects of early design decisions.

Design patterns and differences

Most prototypes share very similar sequence representations. The major outlier is BioJava which relies on a list of symbols to store sequences. The advantages of the choice are not clear when the footprint of the implementation is considered. Some representations are less strictly specified than others. For instance, BioPerl allows residues to be lower or upper case. This results in an increase of the complexity of the client code, which needs either to convert to one case, or to handle cases separately. The case difference is probably required by clients which combine the sequence with a special property (that property would be best described by an annotation).

Defining the Benchmark

This section defines the benchmark that I use to compare the prototypes of sequence representation frameworks.

Data import facility
To date, no major public sequence databank provides sequences as structured data. All databanks rely on files for exchanging data. Also, most bioinformatics tools take sequence files as input and sometimes generate files as output. It is therefore crucial that the framework provide tools for reading common file formats such that data contained in this file can be made available seamlessly to developers who use the framework.

Data export facility
Conversely, an export facility is required such that analysis results such as a small set of sequences can be used with current interactive analysis tools.

Support for types, very large number and size of sequences
Does the sequence representation framework support each kind of biological sequences, namely Proteins, DNA, RNA? What kind of support is being offered to developers who need to manage very long sequences (e.g. chromosome length)? Several bioinformatics softwares incorrectly assume that a sequence can always fit into memory. This causes them to crash when applied to draft sequences of a genome. Other tools cannot handle very large collections of sequences or annotations.

Representation of annotations
The goal of a sequence representation framework is to facilitate the management of the data attached to sequences. During an analysis, a number of annotations can be attached to a sequence. These annotations can be used as data models for visualization tools, which will translate them into a graphics representation. Alternatively, for sequence analysis tasks, they can be used to construct expressions the result of which will filter a set of sequences. Therefore, the representation of annotations, and the ease with which they can be created, attached to a sequence, queried, extended and modified, plays a large role in the usefulness of the framework.

Support for analysis tools
Several analysis tools frequently need to be combined in order to carry out an analysis. The advantage of a canonical representation is most obvious when many tools are interfaced to the framework. These tools can then be used seamlessly to realize the analysis. This is typically achieved by developing a small tool that uses the framework and implement the treatment. Development time should be reduced as compared to file format based approaches, because most of the methods that manipulate the data have been tested during the development of the framework, and by previous analyses.

Support for persistence
Data which persist from one invocation of the program to another is said to be persistent. File systems and database management systems are used to achieve persistence of data. Database management systems are usually preferred for their support for complex queries and transactions (applications that use file systems and grow to a certain point frequently end up implementing query and transaction features from scratch on top of the file system).

Working with persistent instances is required in two cases:
A. For large scale analysis, too many instances cannot fit in memory in the computer allocated to the analysis.
B. When users can interactively change sequence data.

Support for Knowledge Discovery Environments
Knowledge Discovery Environments provide statistics, visualization and machine learning methods in interactive environments. They are very helpful to interpret the results of large scale analysis. Sequence representation frameworks can benefit from an interface with at least one KDE.

Implementation status
What is the status of the implementation of the framework? What language(s) does the framework target? How long has the implementation been in use? What are the nature of the tasks that the framework has been used for?

Testing
How has the implementation been tested? Are regression tests available? Examples that work, etc...

Licensing
How is the framework licensed? This factor will ultimately decide how easily the framework can be reused in development. Is access to the source code granted? Can one easily extend the framework?

Comparison

Data import facility

BSA

Sequences are expected to be provided by CORBA servers. The specification does not impose import facilities to be available. So, at this time, None Available

BioPerl

Comes with SeqIO module to import sequences. Supported formats are: Fasta, Pir, SwissProt, Genbank, SCF, GCG, raw.

BioJava

Comes with IO package to import sequences. Supported formats are: Fasta, EMBL, Genbank

Crover

Comes with imports package. Supported formats: Fasta, PIR, SwissProt, PDB.

Data export facility

None available

Same as import

None available

None available

Support for types, number and size of sequences

Protein/DNA/RNA Large number and size of sequences supported with iterators.

Protein/DNA/RNA Large number of seqs. Supported. Very large sequences probably supported via Ensembl extensions.

Protein/DNA/RNA Heavyweight list of symbols (a symbol is a residue). Large number and size of sequences excluded probably hard to achieve with good efficiency.

Protein/DNA Large number of sequences supported With iterators.

Representation of annotations

As set of individual annotations. Annotation content handled as properties. This probably limits the number of implementations that can be directly accessed through these interfaces. Limited origin are supported (experimental, theoretical).

Annotations are called features. The notion of origin is supported as text.

Annotations are called Features. Each feature can contain features. Features aggregate: Annotation, Location, Source and Type. Annotation stores knowledge as a Map, in a similar way to BSA.

As set of individual annotations. Very similar to BSA, but annotations are classes with full object-oriented features. Annotations can be complex data structures with behaviors.

Support for analysis tools

Specification plans a number of abstract type of analysis.

Blast, HMMER, patterns

Meme, others?

Clustal (clustalnet [12]) PHD HTM, genescan

Support for persistence

Through Persistence CORBA Service.

Through DB and DBI Perl modules. For Ensembl. Limitation: one class one table.

None. Serialization not supported.

Serialization. Problem: no queries. Move to Castor [14] planned.

Support for Knowledge Discovery Environments

N/A

N/A

N/A

A package handles tables and generation files for SGI MineSet.

Implementation status

Two implementations are being developed: EBI openBSA (see ISMB poster) and Netgenics

Implemented and used. See Ensembl project [12]

Limited implementation.

Since spring 1998, in use for RbDe, clustalnet, search for new genes, data mining of biological data

Testing

Too preliminary

Tested during the development. No regression tests.

Tested during the development. No regression tests.

Tested during the development and when new features are introduced (regression tests are applied).

Licensing

Open BSA is Open source

BioPerl is Open source

BioJava is Open source

Documentation is freely available. Packages are licensed.

Remarks

BSA looks promising but the implementation efforts which are ongoing, as well as the support of existing, non compliant implementations, could lead to revisions to this specification.

This is probably your best choice if you are a Perl adept and do not expect too much from the object-oriented design. Be prepared to spend time installing many perl modules.

Probably works well for the applications for which it has been designed.

My personal preference goes here ©. Crover will soon support a subset of BSA. Clustalnet [12] provides an alternative to the BSA analysis scheme.

Conclusions

Several conclusions can be drawn from this comparison.
•The choice of a representation for annotations defines to a large extent the implementation of import services because importing consists of a translation, from one representation (file format) to another (object-oriented representation).
• Extended export services are crucial only for frameworks that rely on files for data storage, as does BioPerl. Other frameworks will rely on persistent storage integrated with a DBMS.
• Representation of the sequence (including annotations) is the most crucial and most difficult part of a framework. It is used by analyses and visualization. Its design will influence the whole framework. The consequences of its design range from import packages to visualization, persistence and data analysis.
A more extended comparison than this poster allows would probably be useful. Including BSA, no publication is available that explains the advantages of one framework design over another. Considering the interest of a framework for building new software and analysis tools, should we accept any design without considering alternatives?

References

- [1] T. Coupaye. Wrapping SRS with CORBA: from textual data to distributed objects. *Bioinformatics*, 15(4), 1999.
- [2] E. Barillot, U. Leser, P. Lijnzaad, and al. A proposal for a standard CORBA interface for genome maps. *Bioinformatics*, 15(4), 1999.
- [3] P.G. Baker, C.A. Goble, S. Bechhofer, N.W. Paton, R. Stevens, and A. Brass. An ontology for bioinformatics applications. *Bioinformatics*, 15(6), 1999.
- [4] P. Karp. An ontology for Biological Function Based on Molecular Interactions. *Bioinformatics*, 2000. In press.
- [5] Object Management Group (OMG). Adopted specification for Biomolecular Sequence Analysis. *lifesci/99-12-01* <http://cgi.omg.org/cgi-bin/doc?lifesci/99-12-01>
- [6] The BioPerl Project is an international association of developers of open source Perl tools for bioinformatics, genomics and life science research. See <http://www.bioperl.org>
- [7] The BioJava Project is an open-source project dedicated to providing Java tools for processing biological data. See <http://www.biojava.org>
- [8] F. Campagne, Conception et développement de systèmes d'aide à la compréhension de données biologiques et moléculaires, Application à la modélisation des protéines G. (Design and Development of systems to help the understanding of biological and molecular data, Application to G Protein-Coupled Receptors). Thèse de doctorat, Université de Nancy I. France.
- [9] F. Campagne, R. Jestin, J.L. Reversat, J.-M. Bernassau, and B. Maigret. Visualisation and integration of G Protein-Coupled Receptor related information help the modelling: Description and Applications of the Viseur Program. *J. Of Computer Aided Molecular Design*, 13(6):625-43, 1999.
- [10] F. Campagne, J.-M. Bernassau, and B. Maigret. The Viseur Program. Laboratoire de Chimie Théorique de Nancy. Viseur Project Home page. <http://transport.physbio.mssm.edu/viseur/viseur.html>.
- [11] F. Campagne and H. Weinstein. Schematic representation of residue-based protein-context dependent data: an application to transmembrane proteins. *J. Of Molec. Graph. & Mod.*, 17(3-4):207-213, 1999.
- [12] F. Campagne. Clustalnet: the joining of Clustal and CORBA. *Bioinformatics*, 2000. In press.
- [13] K. Konvicka, F. Campagne, and H. Weinstein. Interactive construction of Residue-based Diagrams: the RbDe web service. *Prot. Eng.*, Jun;13(6):395-396, 2000.
- [14] The Castor project. <http://castor.exolab.org/>
- [15] Genome Annotation Project Ensembl. <http://ensembl.ebi.ac.uk/>