Data modeling: the key to biological data integration

François Rechenmann✉
Genostar Bioinformatics Solutions, Montbonnot, France

Motivation and Objectives

The advent of NGS technologies is focusing much of the attention towards the data management issue. However, more than their volume, it is the diversity of biological data which constitutes the real bioinformatics bottleneck; a bottleneck which cannot be solved through technological considerations only, such as cloud infrastructures for instance.

A bioinformatics platform must indeed store, organize and give access to a wide span of data and results. First of all, the experimental data and their transformations: not only the sequence data, such as the reads, the assembly files and the resulting contigs - to name the most important ones, but also spectra or metabolic flux measurements. Through the interpretation of these data, biological entities are predicted and characterized: coding regions, regulatory signals, polypeptides, enzymes classes, peptide tags, and so on. All these entities must also be properly described, connected each other and stored in adequately structured data repositories.

Conversely, the programs that implement the analysis algorithms must be able to access these data and these entity descriptions to produce new secondary data and predict new entities.

In this context, conceptual data modeling appears to be the very first task any project aiming at the design and the development of a bioinformatics integrated platform should perform.

Methods

Fortunately, computer scientists have designed a wide range of data modeling tools. Regarding databases, the relational data model provides both a formal well-founded framework, but also leads to efficient implementations as relational database management systems (DBMS). Regarding programs, object-oriented languages, such as Java or C++, allow for the definition of classes and subclasses that can capture the description of the entities these programs deal with.

Moreover, conceptual modeling tools allow designing a data schema independently of its implementation as relations or object hierarchies. Inspired by the entity-relationship model, UML is one of these modeling tools. Formally defined, UML offers a graphical view of a schema that is quite intuitive and may therefore be used during the design phase. Biologists, bioinformaticians and computer engineers can indeed efficiently interact on a shared UML diagram which progressively converges towards a consensual schema.

The Genostar bioinformatics platform for microbial genome annotation and comparison has been designed along these principles. The entities the various software modules had to handle have been identified, together with their relationships. Conceptual differences have been carefully taken into account. As an example, chromosomes, plasmids or segments are subclasses of the class "replicon", while reads and contigs are subclasses of "sequence". When sequences are annotated, features are added onto them. The class "feature" is the superclass of a quite deep hierarchy of classes and subclasses: genes, signals, etc. For example, an object CDS (i.e. coding sequence) is described through a list of variables (or fields), and is connected to the supporting sequence by a relation which is itself described by variables. One of them is the variable "location", which specified where the feature is located on the sequence. This information could not be stored in the feature, nor in the sequence, since a feature may be located on different subsequences, obtained through cut and paste operations from a common sequence.

All the methods provided by the software are also described as classes; the variables of a class are the input and output of the methods. Thus, the type of a variable in such a method description makes reference to the class of entity which can be accepted as values of this variable. This typing mechanism allows the software to check that the input data to a method are consistent with the method description. It also associates useful information to the results the method produces. Such consistency verifications are very useful in a dedicated integrated software platform, which is used by users who wish to concentrate on their analysis process and not on software manipulation.

Results and Discussion

Once the complete data schema has been obtained, it has been implemented in a home-made entity-relationship modeling framework, AROM. This additional modeling level presents several advantages over the direct implementation of the schema as Java classes. It indeed supports query facilities. The software thus offers a large set of built-in queries over the entities and relations on a current workspace. A first example of such a query consists in selecting genes which, according to the computation of their sequence similarity, turned out to be specific to a strain within a set of strains. A second example is retrieving the genes which code for enzymes which catalyze a set of reactions which have been selected on a metabolic map. Moreover, specific queries can be expressed by the user. The software also supports browsing facilities, to explore the connections between the objects. The user can for instance follow the links between a gene, its product, its catalytic functions if any, the reactions it catalyzes, and the metabolic pathways in which these reactions occur.

A nearly identical data schema has been implemented in a relational DBMS. The resulting database MicroB thus integrates genomic, proteic and metabolic data on more than 1500 microorganisms, mainly bacteria at the present time. Since the two schemas, of the software and the database, nearly overlap, data exchanges between these two components are fluid and efficient. Reference data can be extracted from MicroB for comparative analyses in the software; conversely, fully annotated genomes can be stored in MicroB to be later retrieved. SQL queries can be expressed on the contents of the database, but built-in queries together with a dedicated user interface are provided for handling the most standard cases.

The association of the software module dedicated to genome annotation and comparison with the microbial database results in a powerful easy-to-use integrated bioinformatics platform. The explicit representation of objects and their relations offers friendly browsing and querying facilities, helpful type checking, and more generally efficient data management. The user of the platform can concentrate on the data analysis process and forgive all the time consuming and error prone issues resulting from data format conversion and method integration.

But computational biology is a fast evolving scientific domain. New types of data appear, new bioinformatics methods are designed, new methodologies emerge. Again, more than the volume of data, these increasing diversity and complexity appear to be the actual critical issues. Computational biology is a multidisciplinary domain. Multiple interpretations of a concept are frequent and must be resolved when designing software and databases. In this context, explicit and formal data modeling provides very appropriate tools for integrating heterogeneous data, for properly connecting methods and data, and for allowing computer scientists, bioinformaticians and biologists to interact fruitfully over an explicit data schema.

References

1. Peter P. Chen, The Entity-Relationship Model: Toward a Unified View of Data, ACM Transactions on Database Systems, Vol. 1, pp. 9-36, 1976
2. Michel Page et al., Knowledge representation with classes and associations: the AROM system (in French: "Représentation de connaissances au moyen de classes et d'associations : le système AROM "), LMO 2000, Actes des journées Langages et Modèles à Objets, Mont Saint-Hilaire, Québec, Canada; Christophe Dony, Houari A. Sahraoui (Eds.), January 25-28, 2000
3. Technical information on AROM, in English, at http://www.inrialpes.fr/romans/arom/
4. UML: Unified Modeling Language, http://www.uml.org/
5. GenoStar: A Bioinformatics Platform for Exploratory Genomics, François Rechenmann, ERCIM News, No. 51, October 2002, http://www.ercim.eu/publication/Ercim_News/enw51/