NRC as a formal model for expressing bioinformatics workflows

Anna Gambin[1], Jan Hidders[2], Natalia Kwasnikowska[3], Sławomir Lasota[1],
Jacek Sroka[1], Jerzy Tyszkiewicz[1], Jan Van den Bussche[3]
[1]Warsaw University, [2] University of Antwerp,
[3] Limburgs Universitair Centrum

Many *in silico* experiments in bioinformatics can be expressed in the form of workflows where various data-processing steps are connected in a network producing the desired result. Those steps can involve using web-services, invoking software tools, some quick-and-dirty scripts or even manual processing. The process itself is usually data-centered and is designed to analyze large amounts of heterogeneous data, e.g. protein folding prediction or analysis of proteome.

There already are systems developed to assist the researcher in the construction and the execution of such workflows, like Taverna [4] and Kepler [2]. They provide graphical user interfaces for designing workflows, with the possibility of reuse of previously constructed workflows, of storing the result of an execution and even of storing the intermediate results [4].

At present those systems do not provide a formal model for the representation of workflows. The designed workflows are interpreted in an imperative way in the sense that what has been constructed is going to be executed in exactly the specified way. There is a lack of an abstraction level at which the workflow could be analyzed and possibly automatically optimized in specified parts where for instance the order of execution doesn't affect the result.

We propose to employ for that purpose the Nested Relational Calculus (NRC [1]), a well-known formalism from database theory for querying over complex objects. NRC allows the construction of complex data types through the use of nested sets and tuples, and manipulation of data by e.g. iteration over a set, projection of a tuple element, set construction, flattening nested sets and conditional branching.

To represent a bioinformatics workflow in NRC the actual atomic processing steps can be defined as abstract external functions, or "black boxes", embedded into NRC. These black boxes can be e.g. execution of software, a script or a call to a web-service. The system also allows the use of internal functions to hierarchically organize the construction of workflows.

The NRC provides a text based notation for expressing workflows, which is not particularly easy to use. That's why we propose an equivalent graphical notation based on Petri nets.

The same formalism could be used for representing wet-lab experiments with the black boxes encapsulating e.g. the processing of a sample. That could provide the ability to design a workflow incorporating both the data acquisition and data analysis stages of a process. In that case, of course, the wet-lab part of the workflow shouldn't be automatically optimized. The importance of using similar ways for expressing both *in silico* and *in vitro* experiments has also been stressed in the ISXL project [5], although they have defined a new ad-hoc programming language for that purpose.

The use of NRC in the field of bioinformatics is not new. It has already been used as core for the BioKleisli system [3], which facilitates the design and execution of a kind of workflows, although their main concern is data integration and not workflow modeling.

What are the advantages of that kind of formal model? NRC offers methods of workflow optimization, e.g. those parts of a workflow that produce independent results could be replaced by an equivalent workflow producing the same result but that could be executed more efficiently. The Petri net based graphical representation allows us to use methods for analysis of complex workflows, e.g. are there parts of a workflow that can produce a dead-lock, are there branches in the workflow that can never be executed, does the constructed workflow produce a result at all.

The existence of a formal model that combines NRC and Petri nets offers further possibility for developing new techniques for the analysis and optimization of bioinformatics workflows.

References

[1] P. Buneman et al, Principles o programming with complex objects and collection types. *Theoretical Computer Science*, 1995, 149:3–48.

[2] I. Altintas et al, Kepler: An Extensible System for Design and Execution of Scientific Workflows. In the 16th Intl. Conference on Scientific and Statistical Database Management(SSDBM), June 2004.

[3] S. Davidson et al, BioKleisli: A Digital Library for Biomedical Researchers. *Journal of Digital Libraries*, 1997, 1(1):36–53.

[4] T. Oinn et al, Taverna: A tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 2004, 20(17):3045–3054.

[5] A. Tröger et al, A language for comprehensively supporting the in vitro experimental process in silico. *Proc. 4th IEEE International Symposium on BioInformatics and BioEngineering (BIBE 2004)*, March 2004, pp. 47–56.