NRC as a formal model for expressing bioinformatics workflows

A. Gambin 1, J. Hidders 2, N. Kwasiukowska 3, S. Lasota 1, J. Sroka 1, J. Tyszkwiewicz 1, J. Van den Bussche 3

1 Warsaw University, 2 University of Antwerp, 3 Hasselt University

Context
- bioinformatics workflows
  - network of data centered processing steps
- processes involving
  - large amounts of complex data
  - sequence files, BLAST reports
  - XML data
  - a variety of tools
  - EMBOSS suite, BioPerl scripts
  - webservices, Mascot searches

Problems
- workflows execute as a mix of automated scripts and manual intervention
  - difficult to maintain
- results are stored in ad-hoc ways, e.g. files, Excel sheets
  - difficult to manage

Existing solutions
- workflow execution engines
  - Kepler [2], Taverna [3]
  - not based on a formal data model, or too complicated and not data oriented

Our contribution
- using Nested Relational Calculus [1] for modeling data oriented workflows
- many bioinformatics workflows can be modeled in NRC
- advantages of using NRC
  - puts data oriented workflows on a firm foundation
  - formalism is already well understood
- natural approach
  - BioKleisli [4] is also based on NRC

Nested Relational Calculus
- established formalism for querying over complex objects [1]
- complex objects are arbitrarily nested collections and tuples
- collections can be sets, multi-sets and lists
- set-based model: sets () and tuples ()
- typed query language
  - extensible repertoire of base types
    - Boolean, String, Number
    - FASTA sequence file
    - XML, based on a DTD or XML Schema
- complex types: nested sets and tuples

Workflow example – description
- 3D signal maps from LC-MS analysis of blood samples
- two groups: diseased and normal
- extracting clusters corresponding to peptides

Workflow example – data types
- base types
  - String, Number, Boolean, Sample
- complex types
  - input type
    - PatSample = (st: String, sample: Sample, diseased: Boolean)
  - output type
    - FSelCAlg = (txt: String, corr: (CAlg))
    - with CAlg = (dt: PSStats, f: PSStats, svm: PStats)
    - and PSStats = (sensitivity: Number, specificity: Number)
- auxiliary types
  - TestTrain = (test: PepClusters, train: PepClusters)
  - PepClusters = (clustering: Number, path: PathList)
  - PathList = (path: String, diseased: Boolean, intensity: Number)

Workflow example – NRC programs
- top-down design of the workflow
  - after processing and clustering of raw patient data, k-fold cross validation is performed

NRC core operations
- constant value of a base type — “John”, true, 89
- variable of any type, either base type or complex — $\text{PatSample}$
- tuple construction — (name: "John", condition: true, age: 89)
- tuple projection — $\text{PatSample.name}$
- empty set construction — $\emptyset$
- singleton set construction — ($\text{PatSample}$)
- set union — $\text{PatSampleList} = \text{Healthy} \cup \text{Diseased}$
- flattening of a nested set
  - $\text{PatSampleList} = \text{flatten}((\text{Healthy}, \text{Diseased}))$
- iteration over a set
  - for $\text{PatSample}$ in $\text{PatSampleList}$
  - return $\text{PatSample.name}$
- named program definition — $\text{pBLAST}: \text{FASTA} \rightarrow \{\text{AccessionN}\}$
- external programs, used as a "black box"
  - internal programs, help with top-down design
- equality test for base types — $\text{PatSample.name} = "John"$
- emptiness test for sets — $\text{PatSampleList} = \emptyset$
- conditional
  - if $\text{condition then} \emptyset \text{Diseased} \cup \emptyset$ else $\text{Healthy} \cup \emptyset$
- core operations can be combined into programs

- choosing classifiers, defined by a feature selection method
  - $f$-statistic, correlation
- and a classification algorithm
  - Decision Trees (DT), Random Forest (RF), Support Vector Machine (SVM)
- k-fold cross validation to obtain the following performance statistics for each classifier
  - sensitivity, specificity

Acknowledgments
Special thanks to J. Duklowski, A. Gambin, B. Kluge, K. Kowalczyk, J. Tiuryn from Institute of Informatics, Warsaw University, and M. Dadlez from Institute of Biochemistry and Biophysics, Polish Academy of Science, for providing their workflow.

References