A query language for biological networks

Ulf Leser

Bioinformatics, September 2005
Representations of Metabolic Pathways
Representations of Metabolic Pathways
Turning Pictures into Words

- Find all reactions involving a certain substance.
- Find all paths, i.e. chains of reactions, connecting two given substances.
- Find the shortest path between two substances that includes a third substance.
- Given a set of molecules, extract the subgraph which contains all these elements and has the least number of nodes.
Pathway Query Language

- Pathway Query Language (PQL) is a declarative language with syntax similar to SQL (Structured Query Language).
- A PQL query returns a graph, making nesting of queries a possible and expected usage.
Why PQL?

- Talking about a language implicitly forces one to think about the requirements that exist for querying pathways.
- A properly defined language can be used by many pathway databases, reducing the amount of duplicate work.
- A query language acts as an interface between applications and databases. (Allows abstraction.)
- Clear semantics helps to integrate data from heterogeneous repositories.
The basic PQL data model is a graph $G$ with a set of nodes and directed edges. $G$ represents a network of molecules and interactions. A node is either an interaction or a molecule.
The PQL Data Model

- The graph need not be connected.
- Data model similar to those of aMAZE, KEGG, and Reactome.
- The nodes are biological entities or interactions, including properties like type ("gene," "enzyme," "inhibition," "catalysis," etc.) and function (concepts similar to Gene Ontology).
SELECT subgraph-specification
FROM node-variables
WHERE node-condition-set

Example:

```
SELECT *
FROM A, B
WHERE A.name = '3-Isopropylmalate' AND
     B.name = 'EC1.1.1.85'
```
SELECT *
FROM B, C, D
WHERE D.name = ‘L-Lactaldehyde’ AND
B ISA ‘Enzyme’ AND B[-2]D AND
B[-*]C[-*]D AND
C.name = ‘Lactaldehyde’

Fig. 5. Graphical representation of the query given in the text. Dashed lines represent path expressions.
Evaluation of a PQL Query

Underlying database graph

Match graph of the query. Four nodes are matched

Result graph of query. Nodes are added and removed, edges are added.

SELECT B[-1]
FROM A, B, C, D
WHERE A[-1]B[-1]C[-1]D

SELECT B[-1]
FROM A, B, C, D
WHERE A[-1]B[-1]C[-1]D
Networks with Cycles

- Biological networks contain cycles, like feedforward and feedback loops, or reversible reactions.
- PQL doesn’t handle these. PQL only evaluates on cycle-free paths.
  - In cyclical paths, the notion of “all paths” between nodes becomes undefined/infinite.
  - Cycles in larger networks would return too much of the network.
  - Efficiency of computation.
Find all genes whose expression is directly or indirectly affected by a given compound.

SELECT B
FROM A, B
WHERE A.name = 'L-Glutamate' AND A[-*]B AND B ISA 'gene'
Examples

* In the complete set of metabolic reactions, find all feedback loops including a given compound.

* SELECT A[-*]B[-*]A FROM A, B WHERE A.name = 'Methionine' AND A[-*]B[-*]A
The user specifies a set of nodes ...and prompts the system to extract the ... sub-graphs that interconnect each pair of seed nodes via the smallest number of ... links.

FROM A, B, C, D
WHERE A[-*]B[-*]C[-*]D
Find all processes that lead from node A to node B in less than MAX steps and more than MIN steps.


This query fails because it returns all nodes A and B for which there exists at least one path between them longer than M and at least one path shorter than N. Future PQL work?
Find all enzymes for which ATP is an inhibitor.

```
SELECT A
FROM A, B, C, D
WHERE A ISA 'enzyme' AND
    D.name = 'ATP' AND
    B ISA 'reaction' AND
    C ISA 'inhibition'
```
Examples

- Retrieval of all interactions that involve any of a set of molecular species as immediate participant.

- `SELECT A
  FROM A, B
  WHERE A[-2]B`
Examples

* Retrieval of a connected graph that includes a set of specified interactions.

* NOT IN PQL
PQL Implementation

- Oracle Server v9.2
- model for data storage (Node, Edge, Function, Type)
- precomputational procedures for performance
- compiler for PQL queries
- results are returned in two tables to be interpreted by middleware
- two phases: match graph and result graph
- helper tables store all cycle-free paths
- currently 208K paths between 16K nodes and 23K edges from GO
  - these numbers will grow exponentially with larger datasets