Graph Data Management For Biology

Frank Olken

Lawrence Berkeley National Laboratory

presented to

CSB 2003 – Computational Systems & Bioinformatics Conference

Monday, August 11, 2003

Revised Aug. 12, 2003

Copyright Notice

- Copyright 2003 by Univ. of Calif. Regents
- Permission is granted to reuse portions for academic or research purposes, provided portions used are unaltered, and properly attributed.
- Slides may be reprinted for personal use.
- Other users should contact author.

What is Graph Data Management?

- Data is comprised of graphs (labeled, directed)
- Queries are graph based
  - Graph operations, graph pattern matching, ...
- Schemas are also graphs (usually)
- Schema mappings are also graphs (sometimes)

Why Graph Data Management for Biology?

- Biological data represented as graphs
- Biological queries expressed as graph queries
- Graph DBMS can be useful tool
- Avoid reinventing wheel in many applications

Outline of Tutorial

- Introduction
- Graph Terminology
- Applications
- History of Graph DBMSs
- Queries
- Implementation Issues
- Some projects (BGDM, etc.)
- Open Issues / Resources

Intended Audience

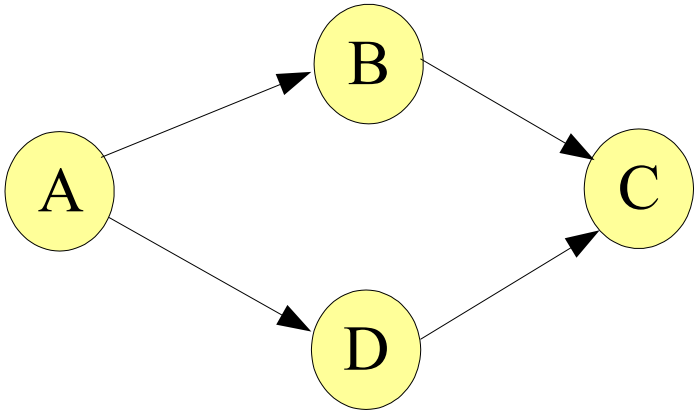
- DBA's
- DBMS designers, implementors, researchers
- Bioinformaticians, computational biologists
- Assumptions:
  - Knowledge of basic relational DBMS technology
  - No graph theory
  - Modest knowledge of biology, chemistry

Why this tutorial?

- Useful material for many bioinformaticists
- We are looking for feedback on these ideas:
  - Use cases, additional query types, criticisms, ..
- We are looking for collaborators, staff
- The field needs more researchers, trained staff

Terminology

Graphs

- Graph = (V, E)
- V = {vertices} = $\{v_i\}$
- E = {edges} =$\{e_j\}$ = $\{ (v_i, v_k) \}$
- 

This is a directed graph.
Undirected graph has no arrows.

Undirected Graphs

- Edges do not indicate direction
- Applications:
  - Protein interaction graphs
  - Contact graphs
  - Schema mappings (some)
  - Matchings
  - Bi-directional chemical reactions

Directed Graphs

- Edges have direction: A--->B
- Applications:
  - Most biopathways representations
  - Lab protocols
  - Genetic maps
  - Taxonomies

Types of Directed Graphs

- Acyclic (DAGs): no cycles
  - Linear (total order)
  - Partial Order (POG)
    - Incomplete ordering
  - Series Parallel Graphs (subclass of POG)
  - Lattices
    - Have top and bottom nodes
- Cyclic Graphs (cycles allowed)

Multi-graphs

- Multiple edges between a pair of nodes
- Used for:
  - Chemical reaction networks (stoichiometry)
  - Chemical structure (bond) graphs
    - Multiple edges for multiple electron bonds
    - Example = benzene ring
- Alternative: label edges with multiplicity

Connectivity

- Connected component:
  - Exists a path between every pair of nodes in CC
  - Used w.r.t. undirected graphs
- Strongly connected component
  - Exists directed paths in both directions between every pair of nodes in SCC

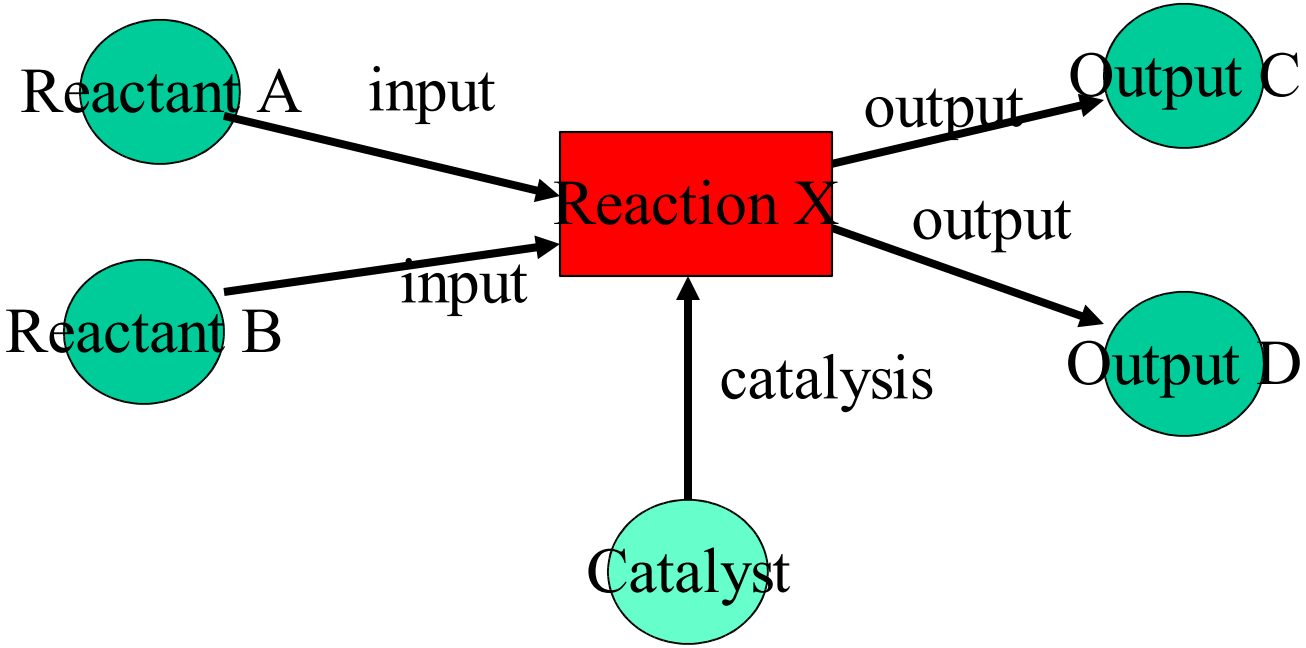
Directed vs. Undirected Graphs

- Biopathways = networks of chemical reactions
  - **also gene expression …**
  - **directed graphs**
  - **nodes = reactions or chemical entities**
  - **edges = relationships (inputs, outputs, catalyst …)**
- Protein interaction networks
  - **undirected graphs**
  - **nodes = proteins**
  - **undirected edges connect interacting proteins**

Bipartite Graphs

- Two sets of nodes
  - e.g., bioprocesses, entities (chemicals, enzymes, ...)
- Edges only connect nodes from different sets
- Applications:
  - Conceptual graphs (relationships, entities)
  - Some biopathways systems (e.g., BP Builder)
  - Contact graph matching

Bipartite Graph representation of chemical reaction



Graphs vs. Hypergraphs

- Graph = vertex set + edge set
  - Edge set = binary relation on vertices
  - Simpler algorithms, less concise
  - Commonly used in biopathways DBs
- Hypergraph = vertex set + edge set
  - Edge set = binary relation on **sets** of vertices
  - More complex algorithms, more concise
  - Used by Ozsoylogu's, Biopax, pathway figures

Graphs vs. Nested Graphs

- Simple graph model = single level graph
- Nested graph model
  - Graphs can be nested inside nodes (Russian dolls)
  - Examples:
    - Hierarchical representation of biopathways
      - Metabolism graph, biopathway graph, reaction graph
    - Chemical structure graphs (bond graph)
    - DNA sequence (linear graph)

Pathway vs. Path

- Pathway:
  - typically a small connected subgraph of a larger graph (e.g., the entire metabolism of a microbe)
  - biological term
- Path:
  - a connected linear graph, i.e., no branches or cycles
  - A---->B---->C----->D
  - a term from graph theory

Adjacency Matrices

- A(i,j) = 1 if there is an edge from node “i” to node “j”
- Undirected graph => symmetric matrix
- Graph composition = boolean matrix multiply
- Typically sparse, inefficient
- Good for dense graphs, transitive closure computations, vector processing, systolic arrays

Stoichiometric Matrix

- Used for chemical reaction networks
- A(i,j) = number of molecules type “i” in reaction “j”
  - Positive A(i,j) = outputs
  - Negative A(i,j) = inputs
- Nonzero elements
  - Correspond to multi-edges in pathway graph

Edge Lists

- Conventional relational representation
- Widely used for many graph algorithms
  - (with indices, sorted/hashed on node labels)
- Tuple A[i,j] = row of relation A
  - Tuple exists if edge from “i” to “j” exists
- Efficient for sparse graphs
- Graph composition = relational equi-join

Graph Data Model vs Relational

- Relational data model = n-ary relations (tables with n columns)
  - Columns are data values, or foreign keys
- Graph data model
  - Edges = binary relations
  - Binary relational model = NIAM

Mapping Relational Data to Graphs

- A row in n-ary relation
- MAPS TO:
  - A single node
  - N edges
  - Note: rows from binary relations (both columns are foreign keys) map to single edges

Applications

Some Applications

- Biopathways, protein interaction networks
- Taxonomies, phylogenetic trees (DAGs)
- Lab protocols (directed graphs)
- Genetic maps (partial order graphs)
- Citation graphs (directed graphs)
- Contact graphs (3D protein structure) (UGs)

More Applications

- Chemical structure (bond) graphs (UGs)
  - Used for chemical information retrieval
- Data Provenance Graphs (directed graphs)
- Hypertext (e.g., web) (directed graphs)
- Interval graphs for shotgun sequence assembly
- Schema representation and integration

Biopathways Databases

- Metabolic pathways
- Signaling pathways
- Gene regulatory networks
- Inferred from:
  - protein interaction networks
  - micro-array data
  - other experiments
- BP = Biopathways

What good is a Biopathways Graph DBMS?

- **Entry point to literature, biopathways data, models, simulation results, ...**
- **Topological framework to hang more detailed mathematical models of biopathways (rate models)**
- **Tool for doing comparative biology**
- **Means of identifying, querying pathway motifs**
- **Graph data model is easier for biologists to use than relational data model of biopathways**

Protein Interaction Databases

- Protein interactions from experiments
- Used to help construction biopathways DB
- Yeast 2 Hybrid Experiments
  - pairwise interactions
- Mass spectrometry experiments
  - indentifies protein complexes
  - NOT pairwise interactions
- PIN = Protein Interaction Network

Biopathways and Protein Interaction Network Database Contents

- **Networks of chemical reactions**
- **Metabolic pathways: bulk reactions, ODE's**
- **Signaling pathwas: rarer reactants, stochastic Petri nets**
- **Gene regulatory networks**
  - **gene expression**
- **Protein Interaction networks**
  - **pairwise protein interactions**
  - **data is very noisy**

Applications of Biopathways Databases

- Formal encoding of biological knowledge
- Drive simulations (add math models)
- Access to literature (links from reactions)
- Access to data (microarray data)
- Assist gene annotation
  - coregulation of genes (e.g., from microarray data) suggests participation in same pathways

What good are biopathways databases? (cont.)

- Comparative analysis of pathways
  - Pathogenicity studies
  - Bioengineering of organisms
  - Phylogeny
- Drug target identification
- Dietary therapy identification
- Drug metabolite identification -
  - possible adverse drug reactions

Comparative Biology

Comparative Biology on Biopathways

- Comparative analyses of DNA/RNA/protein sequences has proven very useful
- Biopathways datasets for many organisms are becoming available
- We can now begin to do comparative analyses of biopathways
- Formal encoding of biopathways needed to permit automated comparisons

Comparative Biology of Sequences

- Sequences are abundant - gigabytes in Genbank
- Sequences are cheap to get now - pennies per base
- Sequences have been collected in databases
- Lots of analysis software available

Comparative Biology of Protein Structures

- Protein structures are expensive to determine - thousands of dollars each
- Few are known - 20K in PDB, 10K distinct
- Some software to compare
- Expensive computations
- Collected in PDB database

Comparative Biology of Biopathways

- Small datasets
  - presently dozens of organisms (mostly microbes)
  - soon hundreds of organisms
  - complete ???
- Expensive to generate
  - often requires micro-array experiments, wet chemistry, gene knockouts, …
- Multiple databases
- Limited analysis software

Requirements for pathway comparative biology

- Data (collections of biopathways)
- Data encoding (graphs)
- Algorithms for graph comparison
- Encodings for graph patterns
- Algorithms for graph pattern matching
- Tools for visualization of graph matchings

Comparative Biology Tools Sequences vs. Pathways

- Global alignment
- Local alignment
- Exact match
- Motif matching
- Dynamic Programming
- String grammar patterns
- Hidden Markov Models
- Phylogeny on sequences

- Graph matching
- Subgraph matching
- Subgraph isomorphism
- Subgraph homorphism
- Approx. graph matching
- Graph grammar patterns
- Graph grammar HMMs ?
- Phylogeny on pathways

Protein Interatction Networks

Protein Interaction Network Databases

- BIND = Biomolecular Interaction DB
- DIP = DB of Interacting Proteins

- These are the two most important, there are others.

BIND=Biomolecular Interaction DB

- **Protein interaction network DB**
- **Pairwise protein interactions = undirected graph**
- **Also other types of data (reactions, …)**
- **http://www.bind.ca**
- **PI: Chris Hogue, Univ. of Toronto**
- **6K interactions, 850 complexes**
- **Includes lots of yeast PI network data**
- **From Yeast 2 Hybrid, and mass spectroscopy expts**
- **ASN.1 import/export**

DIP=Database of Interacting Proteins

- **Protein interactions from experiments**
- **18K protein-protein interactions from 21K experiments**
- **PIs: Ioannis Xenarios and David Eisenberg (UCLA)**
- **Curated**
- **http://dip.doe-mbi.ucla.edu**
- **Available as XML file**
- **Records expt technique, xref to Swiss-Prot, Genbank, PIR**
- **Records binary protein-protein interactions**
- **Graph visualization tool**

Survey Paper on Protein Interaction Databases

- Xenarios, I., and D. Eisenberg, “Protein Interaction Databases”, *Current Opinions in Biotechnology,* vol. 12, pp. 334-339

Biopathways Databases

Biopathways Databases

- Biocyc: Ecocyc, …
- KEGG
- EMP, WIT, …
- Klotho
- aMAZE
- BGDM - Biopathways Graph Data Manager

Biocyc

- Ecocyc, Metacyc, ….
- Developed by Peter Karp (SRI), et al.
- Frame representation, Lisp implementation
- Backend = Oracle, frames=blobs
- A dozen organism groups now use
- Data entry, DB, query, graph drawing
- Primarily metabolic pathways, some signaling
- Separate DB for each organism

Biocyc (cont.)

- Used for E. coli (Ecocyc)
- Extended to other microbes (Metacyc)
- Query by pathway, EC number (of reaction), reactants, citations, …
- Browse ontologies for reactants, enzymes, reactions, ..
- Applied to several other organisms
- http://www.biocyc.org

Biocyc - complex queries

- Biocyc approach to complex queries
  - Read DB into main memory (in Lisp)
  - Write Lisp program for query
  - Run Lisp program on main memory DB

KEGG

- Kyoto Encyclopedia of Genes and Genomes
- PI: Minoru Kanehisa
- Single composite DB for many organisms
- DB, query facilities, pathways drawings
- URL: http://www.genome.ad.jp/kegg
- Select pathway by:
  - EC number, compound number, gene names

EMP, WIT, WIT2

- Eugeni Selkov, Eugeni Selkov, Jr., et al.
- Developed orginally in Russia
- Now at Argonne and Integrated Genomics
- Query pathway by substrate, enzyme, end product, ...
- Servers at IG, Argonne, …
- http://wit.mcs.anl.gov/WIT2

Klotho: Biochemical Compunds Declarative Database

- Developed by Toni Kazic (U. Missouri at Columbia)
- Chemical structure graphs of reactants
- Detailed modeling of chemical reaction mechanisms (cf. stoichiometry only in other DB)
- Written in Prolog
- Public server, DB, open source
- http://www.biocheminfo.org/klotho

aMAZE

- Metabolic + regulatory pathways database
- Developed by EBI in England
- Shoshana Wodak is PI
- Object Oriented DB
- Entity-association model
- Entities: metabolites, proteins, genes, …
- Associations: reactions, catalysis, transport
- 3 tier: presentation, application, storage

aMaze Queries

- Paths: find all paths from A to B
- Pattern search:
  - branch points, feedback loops,
  - pathways affected by a transcription factor
- Pattern discovery (?)
- High level abstraction (?)

History of Graph Databases

Early Graph DBMSs

- Hierarchical DBMSs (trees)
  - e.g., IMS
- Network DBMS (graphs)
  - Various CODASYL databases
  - Lacked declarative query languages
- Semantic Networks (graphs)
  - Common representation for schema, data
  - Evolved into frame systems (used in Biocyc)

Evolution of Graph Databases

- Object Oriented Databases
  - Introduced path queries, path indices, etc.
  - Often used for graph DB (e.g., CAD)

Evolution of Graph DBMSs (IIb)

- Simple graph DBMSs
  - Recursive path queries (Mendelzon)
  - Navigational – Hypertext
- Semi-structured DBMS
  - Graphs
  - Schema Optional
  - Used for documents, hypertext, WWW, federated DBMS

XML, XQuery

- XML is mostly trees (attributes, nested elements)
- Non-hierarchical relationships via ID, IDREF
- DTDs (Document Type Definitions)
  - Structure, very weak type system
- XML Schema Language
  - Both structure and elaborate atomic types
  - Weak description of relationship

RDF

- RDF = Resource Description Framework
- Developed by W3C for metadata, etc.
- Graph data model
- Triples (subject, predicate, object)
  - Subject, predicate (property) are URIs
  - Object is URI or data value
- RDF Schema Language
  - Data type restrictions domains, ranges of predicates
- Used for semantic web (DAML+OIL, OWL)

Biopathways Graph Data Manager

Biopathways Graph Data Manager

- New project at LBNL
- Graph-based data manager
- Graph data model
- Graph queries
- Funded by DOE GTL and DARPA Biospice
- URL:
  - **http://www.lbl.gov/~olken/graphdm/graphdm.htm**

Biopathways Graph Data Model

- Nodes:
  - Chemical entities: metabolites, enzymes, …
  - Bioprocesses: reactions, gene expression, …
- Edges (directed)
  - Indicate relationships
    - input, output, catalyze, inhibit, promote
    - is-a, part-of, element-of, ...
- Recently adopted RDF as graph data model

Taxonomies

- Of enzymes, reactants, organisms, …
- Simplest are hierarchies (trees)
  - each node has exactly one parent (except root)
  - like library classification systems
- Realistic taxonomies are often DAGs
  - directed acyclic graphs (no cycles)
  - nodes may have multiple parents
- Taxonomies specify partial orders

Role of Taxonomies

- Organism, enzyme, metabolite, reaction taxonomies
- Graphs are DAGs (directed acyclic graphs)
- Nodes are concepts
- Directed edges = is-a, instance-of, part-of
- Terms are used to label nodes in query subgraphs
- Generic terms (upper levels of taxonomy) in query subgraphs must be expanded before performing subgraph matching.
- Example: find reaction containing a kinase enzyme

Why are taxonomies DAGs?

- Simple taxonomies are trees …
- However, some enzymes catalyze more than one kind of chemical reactions
- Hence, some enzymes have more than one parent ==> DAG not tree
- Cycles are forbidden in taxonomies

Why are taxonomies DAGs?

- Multi-faceted taxonomies:
  - e.g., Gene Ontology (GO), SNOMED/RT, CPT
  - GO Facets:
    - Molecular function, biological process, cellular component
- Multiple facets ==> taxonomy is not a tree

Why are taxonomies DAGs?

- Conjunctive concepts
- Example:
  - Some bacteria are both aerobic and anaerobic
  - Slime molds are both single cellular and multi-cellular
  - Some proteins may be both structural and catalytic

Context Annotations

- Boolean predicate annotations for biopathway subgraphs
  - Variously: bioprocesses, pathways, ...
- Types: phylogenetic, genotypic, environmental, ...
- Used to indicate when/where bioprocesses occur
- Use binary values (rather than rate models)
- Used as filters in queries

Context for Biological Processes

- Record the context (when, where, why) in which a biological process (reaction, gene expression, etc.) occurs.
- Anatomic context
- Phylogenetic context (which organism)
- Environmental context (pH, O_2, N, ^oC, ...)
- Developmental (cell phase) context

Context for Biological Processes

- Scope of context =
  - Individual bioprocess (e.g., reaction)
  - Biopathway, subgraph of biopathways network
- Specification of context
  - Disjunction of conjunctive clauses, i.e.,
  - $OR_{i=j \text{ to } k}$ ($phase_i$ AND $organism_i$ AND pH_i ...)
  - Can be represented as a graph
  - Contexts can be reified (and named)

Why use context annotations?

- Combine biopathways data from multiple organisms, multiple environments, ...
- Annotate biopathways/reactions with stressors
- Query by context (stressors, environment, organisms) – query contexts used as filters

Specifying Environmental Contexts

- Range conditions over environmental parameters:
  - [NaCl] between 0.0 and 0.5 moles/liter
  - pH between 2 and 4
  - $[O_2]$ greater than 0.01 moles/liter
- Conjunction of simple ranges =
  - Rectangular prism in environmental parameter space
- More elaborate specification would involve linear inequalities (convex polytopes)
  - cf. work by Palsson and Church on linear programming models of metabolic constraints

Why separate context annotations?

- Commonly contexts = attributes of bioprocess nodes
- Separate contexts:
  - Permits reuse of contexts by naming or reference
  - More concise data entry, database
  - Allows different data model, query processing for contexts (e.g., logic DB or constraint DB)
  - Allows more efficient context query processing if contexts are reused

Are contexts graphs?

- Maybe
- Specify context as logical predicate
  - Probably in Description Logic (a restricted logic)
  - Example = OWL (from W3C)
- Encode logical predicate as graph
  - e.g., in RDF
  - cf. similar encodings in conceptual graphs

Graph Queries

Graph Queries

- Paths
- k Shortest Paths
- Graph intersection, union, composition
- Graph Matching
  - subgraph isomorphism
  - subgraph homomorphism
  - subgraph homeomorphism
  - approximate graph matching

More Graph Queries

- Transitive Closure
- Least Common Ancestor
- Largest Common Subgraph

Paths

- Path = connected linear graph
- Example: A--->B--->C--->D
- Length(path) =
  - sum of “lengths of edges” along path
  - typically length of edge = 1

Path Queries

- Fixed length patterns
  - commonplace in Object oriented DB, XML DB
- Regular expressions on paths
  - matching labels on nodes (and edges)
  - recursive, arbitrary length paths
  - see work of Mendelzon, etc.
- Shortest path queries
  - k-shortest paths used a surrogate for most important paths in pathways DB
  - well known algorithms

Path Existence Queries

- Simple directed path existence queries are used for subsumption queries in taxonomies, i.e., does there exist a directed path of is-a edges between 2 concepts
- More general path existence queries allow constraints on path (see previous slide)

Neighborhood Queries

- Neighborhood of radius “r” of subgraph SG of graph G
- Subgraph H of G such that every node in H is within distance “r” of subgraph SG
- distance “r” = length of shortest path
  - edge lengths = 1
- Effect is to include portion of G which is near subgraph SG

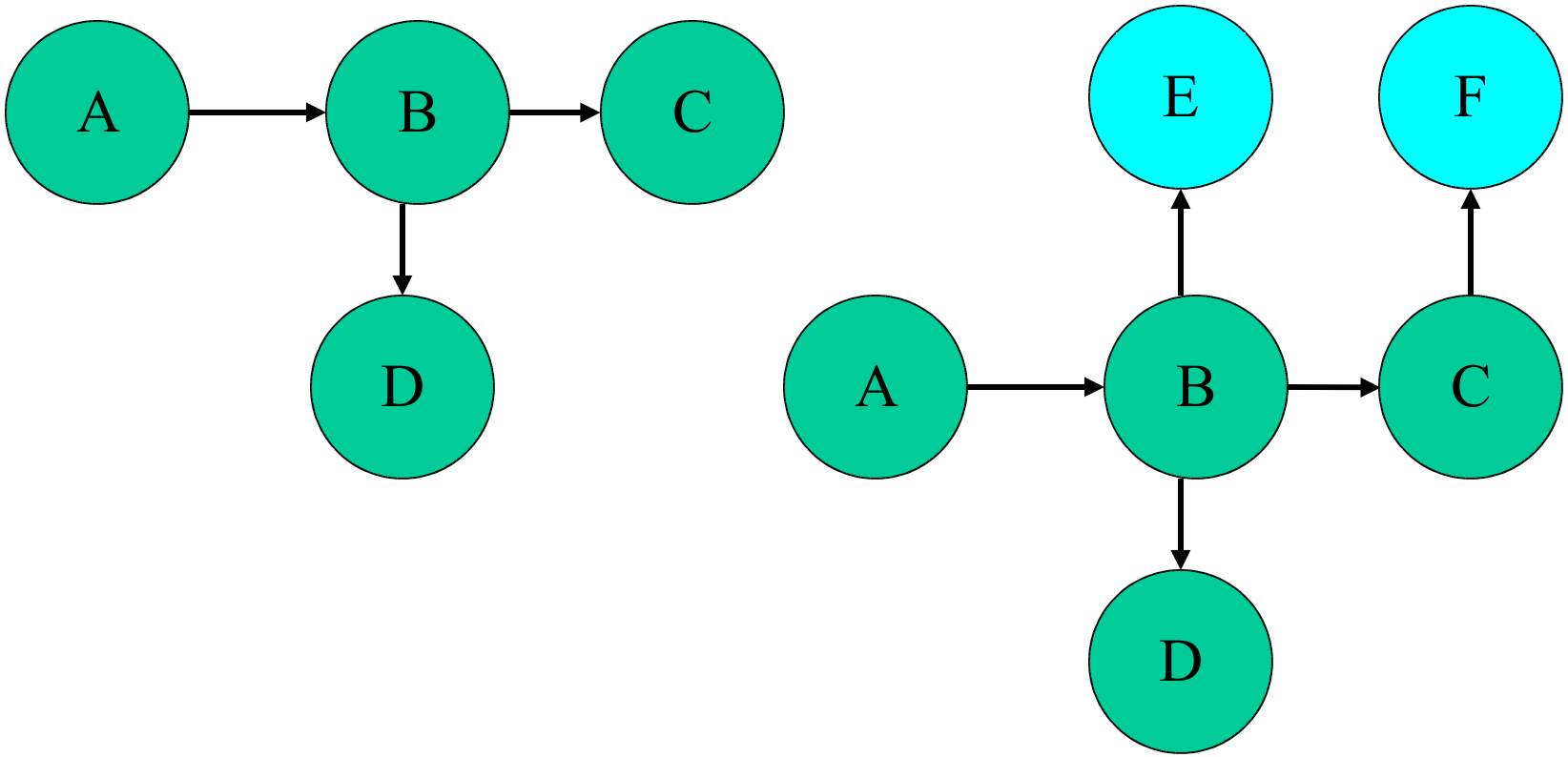
(Sub)graph Matching

- SG isomorphism
  - exact matching of structure and isomorphism on labels (e.g., match labels also if present)
- SG homorphism
  - exact match of structure
  - labels of query graph nodes are generic terms - require expansion via taxonomy graph
- SG homeomorphism
  - SG homomorphism + ellision of some edges

Subgraph isomorphism queries

- Exact match of subgraph structure and labels (sometimes done w/o labels)
- Labels make it easier
- Very common in chemical info retrieval

Example subgraph isomorphism



Subgraph Isomorphism Query Processing

- Top-down backtracking
  - Depth first search (DFS)
  - Breadth first search (BFS)
  - Test for structure match (edge existence)
  - Test for label matching (nodes, edges)
- Bottom-up
  - Use node/edge/path indices to identify fragments
  - Join fragments and/or extend with backtracking

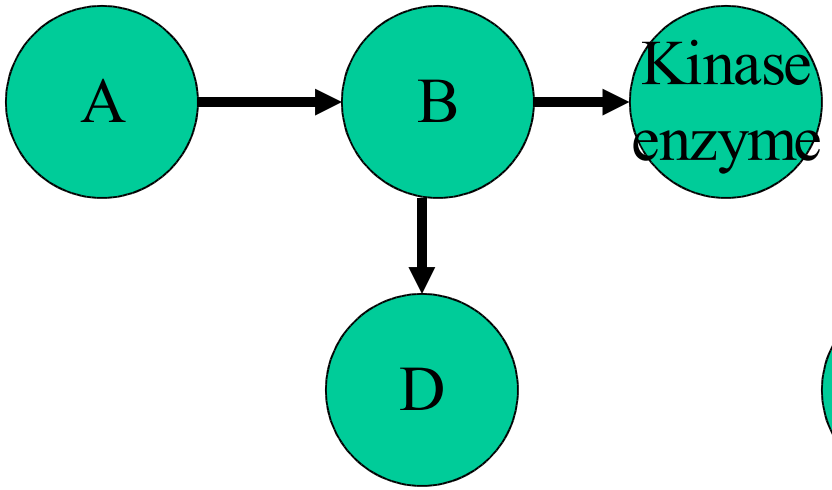
Complexity of SG Isomorphism

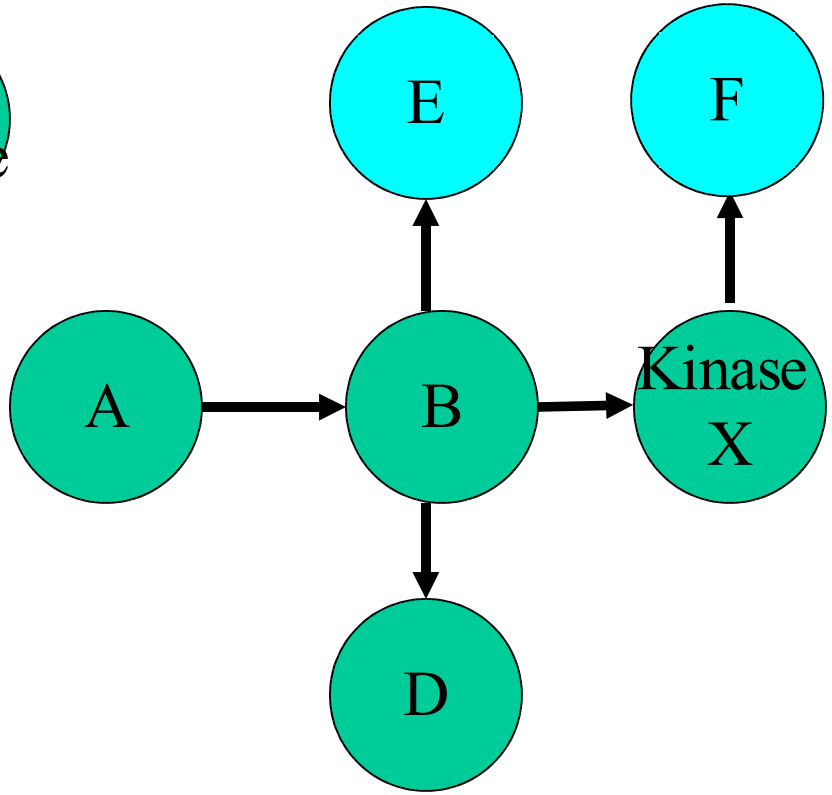
- Depends on sizes/structure of query & DB graphs
- Depends on selectivity of node labels
  - No labels = hard
  - Unique labels = easy
  - Categorical labels
    - Example: atom names in chem. Structure graphs
    - Intermediate complexity
- Depends on fan-out of nodes (edges per node)
  - Bounded fan-out is easier

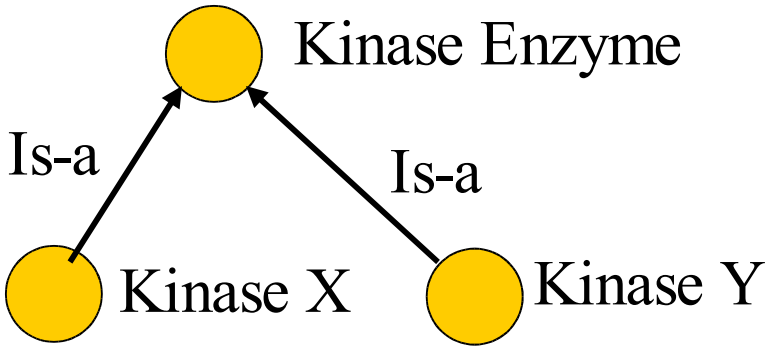
Subgraph homomorphism queries

- Exact match structure (edges)
- Labels on query subgraph are generic terms, e.g., kinase enzyme
- Matching labels are more specific terms subsumed by query node labels, e.g., particular kinase enzymes

Example subgraph homomorphism







Subgraph homorphism query processing

- Subgraph isomorphism testing
- Plus subsumption testing
- Combine via:
  - Joins
  - Backtracking search (nested loop)
- Which to do first?
- Which part of query graph to match first?
- Which node labels to match first?

Subsumption Testing

- In taxonomies (or partonomies)
- Subsumption = directed path is-a edges
- Trees
  - Label leaves consecutively
  - Label nodes with intervals (bottom up)
  - Subsumption = interval containment
  - Subsumption testing in constant time
  - Tree labeling = linear time
  - Extends to mult-faceted classification schemes

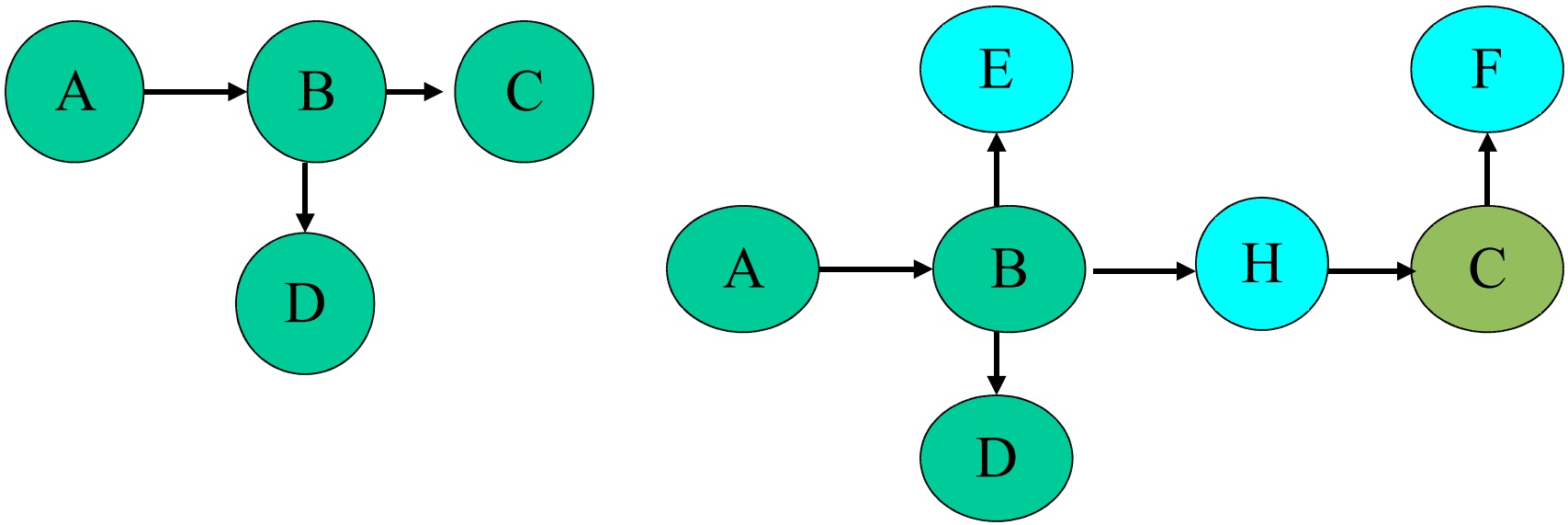
Subsumption Testing (cont.)

- DAGs (directed acyclic graphs)
  - Multiple parents (inheritance)
  - Arises in enzyme taxonomies
  - Can not use tree algorithm
  - Compute transitive closure instead (expensive)
  - TC ==> constant time subsumption testing
  - Various other methods exploit special graph substructures

Subgraph homeomorphism

- Match a subgraph G in a graph H, by performing a subgraph isomorphism test against a “contraction of H”
- Contraction of H = contraction of some edge disjoint paths to single edges

Example subgraph homeomorphism



Note that second graph contains an extra node H between B & C

Approximate Graph Matching

- See work by Dennis Shasha, his students
- Akin to approximate string matching
- Allow: insertion, deletion, substitution of
  - nodes, edges, subgraphs
  - cost for each change
- Dynamic Programming used to find min. cost transformation from graph A to graph B

Node matching

- Needed for (sub)graph matching
- Node match via:
  - exact match of name
  - graph isomorphism of chemical structure graph associated with node
  - approx. string match of sequence (protein, DNA, …)
  - precomputed bipartite matching graph among nodes
  - various algorithmic definitions
  - match node labels + match surrounding context

Boolean Graph Queries

- Graph intersection, union, difference
- Take intersection, union, difference, … of node sets, edge sets
- Note: graph intersection and union can be used to construct majority voting over 3 graphs
- Application: find the difference in metabolisms between two microbes

Majority Graph Query

- Majority (A, B, C) =
- (A&B) U (A&C) U (B&C)
- where
  - & = graph intersection
  - U = graph union
- Usage: to combine multiple (unreliable) protein interaction graphs
- Can be extended to other voting queries

Graph Composition

- Compose two graphs A and B
- Connect outputs of graph A to inputs of graph B
- Used to construct pathways from individual reactions
- Also used to connect pathways, metabolism of co-existing organisms, ...

Shortest Path Queries

- Identify “important” paths from
  - nutrients, or intermediate products
  - to chemical outputs
- Shortest paths queries are attempt to generate most important pathways

Transitive Closure Queries

- Find all products ultimately derived from a particular reaction
- These are potentially affected by knockout (or defect) of root gene
- Find all descendants of node in tree (e.g., rooted phylogenetic tree, taxonomy, ...)
- A-->B, B-->C, transitive closure includes A-->C

Transitive Reduction Queries

- Inverse of transitive closure
- TR(G) = minimal G' such that TC(G')=G
- Graph
  - A--->B, B--->C, A-->C
- Transitive Reduction of graph
  - A-->B, B-->C
- Used in shotgun sequence assembly

Transitive Reduction of Interval Graph

- Interval graph:
  - Node per interval (on real line)
  - Edge if intervals overlap
- Transitive Reduction of Interval Graph
  - Is the partial order graph of the edges
  - Total order if no edges are contained in one another
- Arises in shotgun sequence assembly

Least Common Ancestor Queries

- Find the closest ancestor common to sereval nodes in a directed graph
- Observe multiple products are co-regulated
- Identify putative master control reaction
- Classically defined on trees (or DAGs)

Graph Decomposition Queries

- Decompose a large graph into smaller components:
  - e.g., with minimal interconnectivity
  - Identify distinct subsystems
  - Components also called “modules”
- See paper by Wolf and Arkin

Frequent Subgraph Recognition

- Data mining for frequent subgraphs
- Used to find biopathway graph motifs
- Analog to motif mining in DNA sequences
- Examples:
  - Inverters, oscillators, NAND gates, amplifiers, flip-flops, cascades, ...
- See paper by Wolf and Arkin

Visual Query Languages

- Similar to a drawing program (or CAD editor), or pathway editor
- Main Panel:
  - Query graph drawing area
  - Also used for selection from side panels (taxonomies, etc.)
- Side Panels (pop up menus (graphs?) for browsing, selection):
  - Contexts
  - Taxonomies: enzymes, chemicals, organisms, ...
  - Partonomies: anatomy, ...
- Tool panel: for drawing query graphs, predicate specification

Visual Graph Query Tool for Biopathways

Contexts

Taxonomies

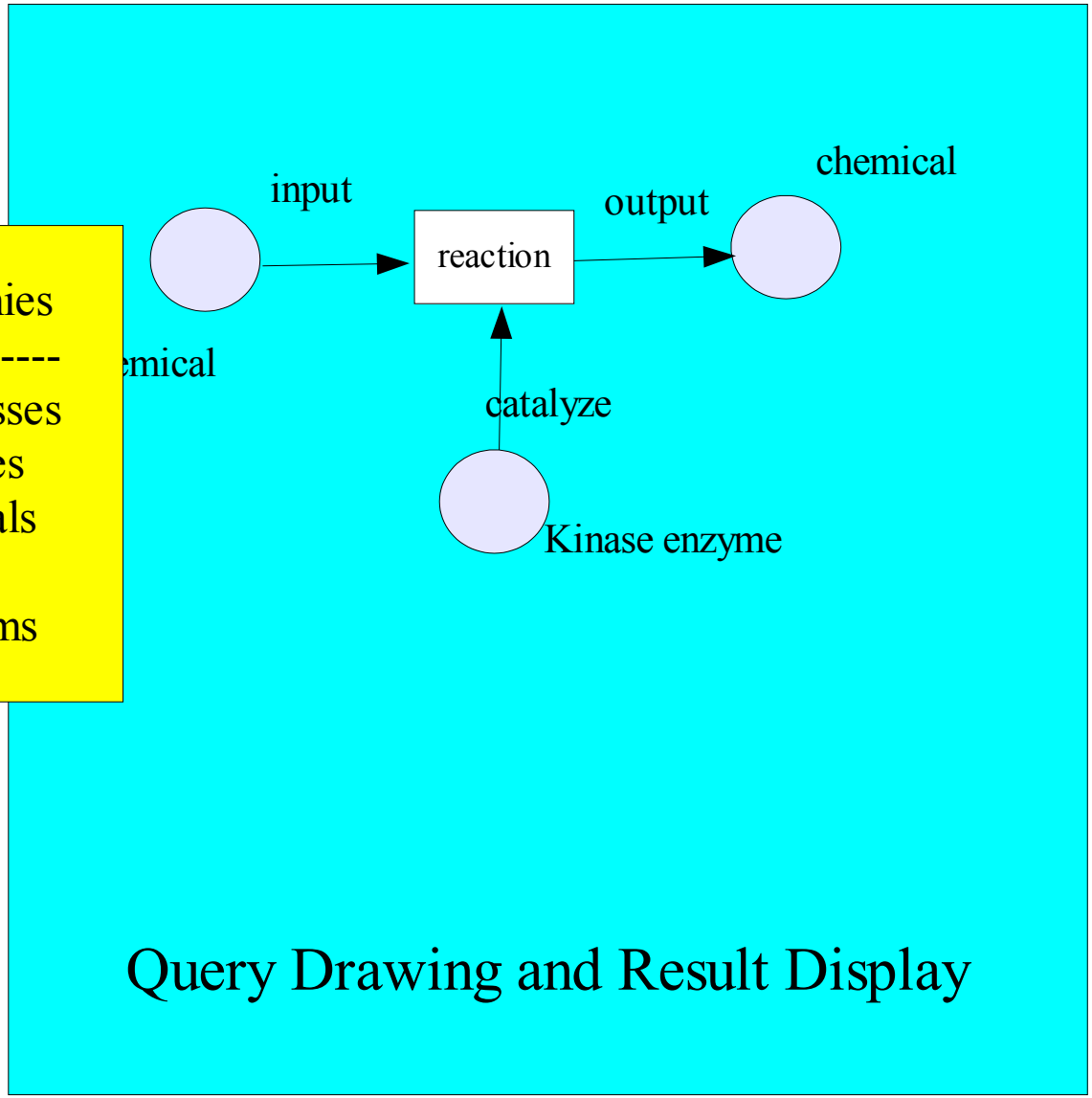
Partonomies

Drawing Tools
==========
Nodes
Edges
Ellipses
Labels
Operators

Save Query
Run Query
Save Result

Taxonomies

Bioprocesses
Enzymes
Chemicals
Genes
Organisms



Graph-based Visual Query Tools

- Graphlog (A. Mendelzon, M. Consens)
- ER Query Tools (commercial)
- Case Western Biopathway DB (Ozsoyoglu's)
- AMAZE (?)
- Others

Implementation Issues

Implementation Issues

- Layered vs. Integrated Architecture
- Backend Storage (DBMS) System
- Main memory vs. Disk Query Processing
- Indices
- Query Execution Strategies
- Query Optimization Issues

Layered Architecture

- Graph DBMS atop Storage/DBMS backend
- Advantages:
  - Graph algoritms code outside DBMS
  - Easier implementation
  - Portability across backends
  - Flexibility in choice of query language
  - Flexibility in choice of implementation language

Layered Architecture (cont.)

- Disadvantages
  - Requires reimplementation of much of query optimizer, engine
  - Hard to integrate graph QL with underlying QL
  - Performance problems: many calls across backend interface
  - Cache management issues
  - Adopted by many researchers:
    - BGDM (Olken), RQL, Ozoyoglu's (?), early GIS systems, LDL++, ...

Integrated Architecture

- Graph DBMS functions embedded in a DBMS
- Advantages:
  - Avoids much reimplementation of query opt, query engine
  - Uses existing cache management code
  - Can exploit existing index facilities
  - Fewer calls across backend interface
  - Better opportunities for query optimization

Integrated Architecture

- Disadvantages
  - No portability across backends
  - Possible problems with federated graph DB
  - Requires coding with DBMS
  - More difficult implementation
  - Little choice of implementation language ©
- Adopted by:
  - DB vendors: IBM (?), Oracle (?)

Backend Storage System

- Object Relational DBMS
- Object Oriented DBMS
- Transactional Storage Server

Object Relational DBMS Backend

- Examples: DB2, Oracle, PostgreSQL
- Advantages:
  - Transaction Management
  - Relational query capability
  - Some recursive queries (transitive closure)
  - Easy integration of legacy databases
  - Scalability, Parallel query evaluation (DB2, Oracle)
  - Easier federated DB implementation

Object Oriented DBMS Backend

- Easier integration of graph algorithms with backend DBMS (e.g., C++)
- Disadvantages:
  - No open source OODBMS's
  - Scalability, parallelism ?
  - Harder to do federated DB
  - (Usually) weaker query language facilities, optimization

Transactional Storage Server Backend

- Example: Berkeley DB from Sleepycat
- Provides:
  - transaction management,
  - indexing,
  - selection queries
- Advantages:
  - Open source, potentially high performance

Main Memory Query Processing

- Advantages:
  - Easier implementation, more flexible graph algorithms
  - Possibly higher performance due to faster memory access (1 usec vs. 10 ms.)
- Disadvantages:
  - Difficulty in scaling up DB size
  - Constrained to a few GB, perhaps 10's of GB

Disk Based Query Processing

- Advantages:
  - Uses existing index structures, query engines
  - Lots of parallelism available
  - Scales to very large databases: terabytes
- Disadvantages:
  - Constrained, difficult implementation of graph algorithms
  - Performance problems due to slow disk accesses

Indices

- Nodes (by label)
- Edges (by label, type, nodes, node labels)
- Paths (sequence of node, edge, node, ...)
  - All paths of length k
  - All paths
  - Prefix, suffix, subpath matching
- Transitive Closure Indices
  - Use with taxonomies to speed subsumption tests

Query Optimization

- Top vs. bottom up query processing
- Depth first vs. breadth first searching
- Choice of algorithms
- Ordering/interleaving of evaluation query fragments
- Index usage
- Main memory vs. disk-based
  - Hybrid pushes selections/projection into backend DBMS, more complex query processing in main memory
  -

Query Optimization Factors

- Query structure
- Database graph structure, statistics
  - Node fan-out, predicate selectivity, connectivity, diameter, type of graph (cyclic, DAG, tree)
- Index availability, graph data structure
- CPU (speed, number, hyperthreading)
- Memory (speed, size, caching, interleaving)
- Disks (number, layout, access time, transfer rate)

Top Down Query Processing

- a.k.a. Tuple substitution method in RDBMS
- Basis of nested loop join query algorithm
- Relies on backtracking
- Used in many graph algorithms
- Often difficult to parallelize – esp. depth first search
- Widely used in Prolog implementations

Bottom Up Query Processing

- Construct algebraic query plan (tree)
- Leaves are base relations
- Internal nodes are operators (join, select)
- Widely used in relational DBMS's
  - With sort-merge or hash joins
- Many possibilities for parallelism, pipelining
- Hard to formulate many graph algorithms in this form

Graph DM Systems

- RQL
- BGDM

RQL

- Christopher Vlassides (Univ. of Crete)
- FORTH Project
- Query language for RDF Databases
- Allows graph pattern matching
- Translates queries into SQL
- Uses PostgresQL as backend

Architecture of BDGM

- 3 processes
  - applications, graph data manager, DBMS
- Persistent store = relational DBMS
- Graph query processing in main memory
- Applications programs invoke BGDM via
  - SOAP, XML data exchange
- Applications: pathway viz, editor, analysis
  - (not included)

BGDM cont.

- Still in design phase
- Data sources:
  - **Arkin Lab, VIMSS, Synechococcus, et al.**
  - **Various public biopathways databases**
- BGDM will be open source software
- URL:
  - **http://www.lbl.gov/~olken/graphdm/graphdm.htm**

Some Open Issues

- Data models: RDF, nested, hypergraphs
- Query language design
- Visual query language
- Query plan representation
- Scalable graph algorithms
- Implementation approaches
- Query optimization
- Parallelism
-

More Open Issues

- Are GDMS useful – or should this functionality be in applications programs?
- Do integrated GDMS/RDBMS make sense?
- Extensibility
- Can we build a common GDMS – or are applications too specialized?
- Relationship of GDMS vs. Logic DBMS

Conclusions

- Graph DM is key technology for bioinformatics
- GDMS are doable and useful for many bioinformatics applications
- General purpose GDMS are doable
- GP GDMS can be widely reused
- Many open research topics

Funding Acknowledgements

- Partial funding from VIMSS Genomes to Life Project at LBNL (PI = A. Arkin). Funded by Dept. of Energy Genomes to Life program.
- Partial funding from LBNL Biospice Project, (PI = A. Arkin). Funded by DARPA Biocomp program.
- Partial funding from Synechococcus GTL project, via subcontract from Sandia. Funded by Dept. of Energy Genomes to Life program. (PI = G. Heffelfinger)

Personal Acknowledgements

- **Vicky Markstein (inSilico) – encouraged tutorial**
- **Ying Xu (ORNL) – encouraged tutorial**
- **Manfred Zorn (LBNL) - assisted with some figures**
- **Sylvia Spengler (NSF) - encouraged work on graph databases**
- **Adam Arkin (LBNL) - supports graph DB work**
- **Kevin D. Keck (LBNL) – BGDM staff, many discussions**
- **Denise Wolf (LBNL) – biochemical network analysis**
- **Paul Brown (IBM) – discussions on implementation of GDMS**
- **M. Fernandez, J. Simeone, P. Wadler - Xquery, functional programming approaches to query languages**
- **Meral Ozsoyoglu – comments on tutorial, references**

Contact Information

- Frank Olken
  - **Lawrence Berkeley National Laboratory**
  - **Computational Sciences Research Div.**
  - **1 Cyclotron Road, MS 50B3238**
  - **Berkeley, CA 94720-8147**
  - **http://www.lbl.gov/~olken**
  - **http://www.lbl.gov/~olken/graphdm/graphdm.htm**
  - **Email: olken@lbl.gov**
  - **Tel: 510-486-5891**

Resources

Conferences on Graph DB

- SIGMOD, PODS, VLDB
- Graph Grammar Conf.
- Graph Transformation Conf.
- IEEE Conf. On Visual Languages
- WWW Conf.
- Semantic Web Conf.
- ISMB

Journals on Biopathways DB's

- Nucleic Acids Research – first issue annually
- Bioinformatics
- Journal of Computational Biology

Conferences on Biopathways DB's

- Intelligent Systems for Molecular Biology (ISMB)
- IEEE Computer Society Bioinformatics Conference

Web Pages on Biopathways DB's

- Biopathway Graph Data Manager Web Page
  - http://www.lbl.gov/~olken/graphdm/graphdm.htm
  - See numerous links within above web page
- KEGG Links to Biopathways and Other Databases
  - http://www.genome.ad.jp/kegg/kegg4.html

Survey Papers on Biopathways

- Denise M Wolf and Adam P Arkin, “Motifs, modules and games in bacteria”, *Current Opinion in Microbiology*, 2003, vol. 6, pp. 125-134.
- Eric Alm and Adam P. Arkin, “Biological Networks”, *Current Opinion in Structural Biology*, 2003, vol. 13, pp. 193-202.