CSCI2950-C
Interaction Networks
Lecture 10

Ben Raphael
November 3, 2009
http://cs.brown.edu/courses/csci2950-c/

Outline

• Network alignment and querying
• Color coding and randomized algorithms.
Protein Interaction Networks

(Over?)simplify interactions between proteins as a binary, static relationship.

Protein-Protein Interaction network
– an undirected graph (usually)
  • Nodes: protein
  • Edges: interactions
  • Edges may have weights indicating confidence.
– Yeast DIP network: ~5K proteins, ~18K interactions
– Fly DIP network: ~7K proteins, ~20K interactions.
– Human. ~20K protein. ~50K interactions.

PathBLAST: Computational Formulation

• $I = \{\text{start vertices}\}$, e.g. receptors.
• Goal: Find highest scoring paths $I \rightarrow v$ for all $v$ in $G$. 

Scott, et al. JCB 2006
PathBLAST: Computational Formulation

- **Given:**
  - Undirected weighted graph
    \( G = (V, E, w) \)
  - Set of start vertices \( I \) and end vertex \( v \),
- **Find:** a minimum-weight simple path \( P = (v_1, e_1, v_2, e_2, \ldots, e_{k-1}, v_k) \) starting in \( I \) and ending at \( v \):
  - \( v_i \) in \( I \) and \( v_k = v \).
- Recall: Simple path \( v_i \neq v_j \) if \( i \neq j \)
- NP-hard in general (reduction from TSP)
- Let \( w_k(v) = \text{weight of above.} \)
  - Dynamic programming solution (whiteboard)

Scott, et al. JCB 2006

Color-coding
(Alon, Yuster, & Zwick)

- Assign each vertex random color between 1 and \( k \).
- Colorful path: path w/ distinct colors.
- Colorful path \( \rightarrow \) simple path.
- **Goal:** find colorful paths
  - Dynamic programming solution (whiteboard)
  - High-scoring path not discovered when two vertices have same color.
  - Repeat for many random colorings. (How many?)
Adding extra constraints

- Require a protein: assign it a unique color.
- Require a specific number of proteins from a set $T$: $W(v, S, c) = \min$ weight of path ... (same as above) and contains exactly $c$ vertices in $T$.
- Order constraint on proteins in path
  - Membrane proteins $\rightarrow$ transcription factors.

Find: min-weight binary tree
with $k$ nodes
Rooted at $v$.
Every leaf is in $I$.

\[
W(v, S) = \min \left\{ \min_{u: c(u) \in S \setminus \{c(v)\}} W(u, S \setminus \{c(v)\}) + w(u, v), \right.
\]
\[
\min_{(S_1, S_2): S_1 \cap S_2 = \{c(v)\}, S_1 \cup S_2 = S} W(v, S_1) + W(v, S_2) \right\}
\]
Color-coding
(Alon, Yuster, & Zwick)

• Extends to many other cases of subgraph isomorphism problem:
  Does a graph G have a subgraph isomorphic to graph H?
• H = simple path of length $k$.
• H = simple cycle of length $k$.
• H = tree.
• H = graph of fixed (bounded) tree-width

Network Querying Problem

• Species A
  • well studied
  • protein interaction sub-networks defined by extensive experimentation
• Species B
  • less studied
  • little knowledge of sub-networks
  • protein interaction network known using high-throughput technologies
• Can we use the knowledge of A to discover corresponding sub-networks in B if it is “present”?
Graph Isomorphism

$G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic provided:

There is a bijection between the vertices $V_1$ and $V_2$ that preserves edges

i.e.

There is a 1-1, onto function $\Phi: V_1 \rightarrow V_2$ such that

$(u,v) \in E_1$ if and only if $(\Phi(u), \Phi(v)) \in E_2$

Graph Isomorphism Problem

Are $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ isomorphic?

Neither known to be in P or to be NP-complete.
Subgraph Isomorphism Problem

Is \( G_1 = (V_1, E_1) \) isomorphic to a subgraph of \( G_2 \)?

NP-complete problem.

Network Querying Problem

- Given a query graph \( Q \) and a network \( G \), find the sub-network of \( G \) that is
  - Isomorphic to \( Q \)
  - aligned with maximal score
- NP-complete: subgraph isomorphism.
Network Querying Problem: Homeomorphic Alignment

Match of homologous proteins and deletion/insertion of degree-2 nodes

Graph Subdivision and Homeomorphism

Subdivision of an edge: “insert” a vertex.

Subdivision of G is graph obtained by subdividing some edges.

G and G’ are homeomorphic provided there is an isomorphism from a subdivision of G to a subdivision of G’
Network Querying Problem: Score of Alignment

$$\text{Score} = \sum h(q,v) + \delta_d(\# \text{Del}) + \delta_i(\# \text{Ins}) + \sum w(v_i, v_j)$$

- **Score**
  - Sequence similarity score for matches
  - Penalty for deletions & insertions
  - Interaction reliabilities score

Network Querying Problem

- Given a query graph $Q$ and a network $G$, find the sub-network of $G$ that is
  - homeomorphic to $Q$
  - aligned with maximal score
Complexity

- Network querying problem is NP-complete. (for general n and k)
  - by reduction from sub-graph isomorphism problem

- Naïve algorithm has $O(n^k)$ complexity
  - $n =$ size of the PPI network, $k =$ size of the query
  - Intractable for realistic values of $n$ and $k$
  - $n \sim 5000, k \sim 10$

- We use randomized “color coding” technique developed by [Alon et al, JACM, 1995] to find a tractable solution.
  - Reduces $O(n^k)$ to $n^{2^{O(k)}}$.

QNET

- Implemented for tree-like queries.

- Color coding approach to search for the global optimal sub-network.

- Extension of QPATH [Shlomi et al., 2006]
  - Solves the problem of querying chains using color coding approach.
Color Coded Querying - Trees

Query has $k$ nodes.

Randomly color the network with $k$ distinct colors. Suppose optimal sub-network is “colorful”. Use the colors to remember the visited nodes.
DP solution for Color Coded Querying - Trees

Example

FIG. 1. (a) An example of a tree query graph and the corresponding alignment subgraph. Numbers on the query graph’s edges represent an arbitrary ordering of children nodes. Aligned query nodes and graph vertices are connected with dashed lines. Nodes in the skeleton graphs appear in gray. (b) A simulation of the dynamic programming recursions. For simplicity, we denote color sets as {} . Matched vertices are awarded by +5, insertions and deletions are penalized by −3 and edge weights are as shown.
Probability of failure

- The optimal alignment can be found only if the optimal sub-network is “colorful”.

\[ P(\text{failure}) = 1 - \frac{k!}{k^k} \leq 1 - e^{-k} \]

- Repeat color-coded search multiple times until probability of failure \( \leq \varepsilon \).

Number of Repeats

- Necessary number of repeats to guarantee a failure \( \leq \varepsilon \)?

  - Repeat \( N = (\ln 1/\varepsilon) e^k \) times, then

\[
\begin{align*}
P(\text{failure per trial}) &\leq 1 - e^{-k} \leq e^{-e^k} \\
P(\text{failure in all trials}) &\leq e^{-e^k N} = e^{-\ln (1/\varepsilon)} \leq \varepsilon
\end{align*}
\]
Restricted Color Coding

Idea: take advantage of queries whose proteins tend to have non-overlapping sets of homologs.

Special case: no insertions allowed, and query nodes have non-overlapping sets of homolog, then \( P(\text{failure per trial}) = 0. \)

Network Querying with Color Coding

Approach

Repeat \( N \) times:

1. Randomly color the network graph.
2. Query the network graph for a high scoring subnetwork.
3. Use the DP algorithm to analyze the query.
Querying General Graphs

• Extend algorithm to general query graphs.

• Idea:
  – Map the original graph into a tree, i.e. tree decomposition.
  – Solve the querying problem on this tree using DP.

Color Coded Querying – General Graphs

Map the original query into a tree using tree-decomposition.

\[ G \rightarrow T \]

node=set of vertices

vertex
Tree Decomposition

Given $G = (V, E)$.
Form tree $T = (X, E_T)$
Each $X_i \subseteq X$ is a subset of $V$.
For all edges $(u,v) \in E$: there is a set $X_i$ containing both $u$ and $v$.
For every $v \in V$: the nodes that contain $v$ form a connected subtree.

Color Coded Querying – General Graphs

Tree decomposition is not unique.
Width of a tree decomposition is the size of its largest node minus one: $\max_{e \in T} |X_i| - 1$
Treewidth of a graph $G$ is the minimum width among all possible tree decompositions of $G$. 
Color Coded Querying – General Graphs

The treewidth of a graph $G$ is the minimum width among all possible tree decompositions of $G$. DPs on trees can usually be extended to tree decompositions. Problems solved efficiently on trees by DP can be solved efficiently on graphs with bounded treewidth.

**Tree decomposition**

- A tree decomposition:
  - Tree with a vertex set associated to every node.
  - For all edges $(v,w)$: there is a set containing both $v$ and $w$.
  - For every $v$: the nodes that contain $v$ form a connected subtree.

Tree decomposition

- A tree decomposition:
  - Tree with a vertex set associated to every node.
  - For all edges \( \{v, w\} \): there is a set containing both \( v \) and \( w \).
  - For every \( v \): the nodes that contain \( v \) form a connected subtree.

Treewidth (definition)

- **Width** of tree decomposition:
  \[ \max_{i \in I} |X_i| - 1 \]
- **Treewidth** of graph \( G \):
  \( \text{tw}(G) = \) minimum width over all tree decompositions of \( G \).
Color Coded Querying – General Graphs

Original query has k nodes and tree-width t.
Randomly color the network with k distinct colors.
Running time

- \( n \) = size of network, \( k \) = size of query.

- Tree queries:
  - Reduces \( O(n^k) \) to \( n^2 2^{O(k)} \).
    - Tractable for realistic values of \( n \) and \( k \).
    - \( n \sim 5000, k \sim 10 \)

- Bounded-tree-width graphs:
  - \( t \) : tree-width
  - \( n^{(t+1)2^{O(k)}} \)

Heuristic for Color Coded Querying - General Graphs

1. Extract several spanning trees from the original query.
Heuristic for Color Coded Querying - General Graphs

1. Extract several spanning trees from the original query.
2. Query each spanning tree in the network.
1. Extract several spanning trees from the original query.
2. Query each spanning tree in the network.

Heuristic for Color Coded Querying - General Graphs

1. Extract several spanning trees from the original query.
2. Query each spanning tree in the network.
3. Merge the matching trees to obtain matching graph.
Test: Cross-species comparison of MAPK pathways

- Query: human MAPK pathway involved in cell proliferation and differentiation.
- Network: fly PPI network
- Result: a known fly MAPK pathway involved in dorsal pattern formation.

Test: Cross-species comparison of protein complexes

- Queries: trees of size 3-8 extracted from ~100 yeast hand-curated MIPS complexes.
- Network: fly PPI network
- Result:
  - ~40 of the queries resulted in a match with >1 protein.
  - 72% of the matches are functionally enriched. (pvalue < 0.05)
    - 17% of the random trees extracted from network are functionally enriched.
Additional Problems

1. Find conserved subgraphs
   *Heavy* subgraphs in product graph
2. Multiple network alignment

Sources

Sources

- QNET slides: modified from slides of Banu Dost.