Identification of Novel Protein Complexes through Pseudo Clique Enumeration

Andrew Schoenrock

December 3, 2010

Outline

- Introduction & Background
    - Protein-Protein Interactions and Protein Complexes
    - PIPE: Protein-protein Interaction Prediction Engine
    - What's the problem?
- Generating Discrete Objects with Reverse Search
- Pseudo Clique Enumeration based on Reverse Search
    - Basic Definitions
    - Algorithm
    - Example
- Identifying Novel Human Protein Complexes
- Conclusions

Introduction: Proteins & Protein-Protein Interactions

- Proteins are essential organic compounds in all organisms and participate in virtually every process within a cell
- Proteins can work together (interact) to carry out various functions and do so for a majority of biological functions
- Protien-protein interactions are responsible for a cell's general behaviour and it's response to stimuli
- A protein complex is a group of two or more that interact with one another to perform a certain function
- Protein complexes are a cornerstone of many biological processes

Introduction: PIPE

- PIPE: Protein-protein Interaction Prediction Engine
- A computational tool used to predict whether two proteins interact or not
- PIPE3 has produced the first proteome-wide protein-protein interaction predictions for *C. Elegans* and *Homo Sapiens* organisms
- These proteome-wide predictions can be viewed as graphs where:
    - Each vertex represents a protein
    - Each edge represents an interaction (known or predicted)

What is the problem?

- Problem: Enumerate all protein complexes within the proteome-wide interaction prediction graphs to identify previously unknown protein complexes.

- What will a protein complex look like in the graph?
  - Protein complexes are identified as dense subgraphs (pseudo cliques) where each protein interacts with a significant number of the other complex proteins.

- Base Problem: Enumerate all dense subgraphs G' of a graph G such that the G' has a significantly high number of edges.

Reverse Search for Enumeration

- Several known search techniques for enumeration problems
  - backtrack search
  - incremental search
  - DFS or BFS when objects to be listed are vertices of a graph

- Reverse search is an exhaustive search technique which can be considered as a special graph search

Reverse Search for Enumeration

- Assume we have a problem for which we would like to enumerate a set of objects
- Let G be a graph where the vertices represent the objects we wish to enumerate and the edges represent two objects that are considered adjacent
- A local search algorithm on G is a procedure to move from one vertex to a larger neighbour with respect to some objective function
- A vertex without a larger neighbour is a local optimum

Reverse Search for Enumeration

Imagine a simple case where there is only one local optimal vertex v^*.

- Consider the digraph T with the same vertex set as G and the edge set made up of the ordered pairs (x, x') of consecutive pairs generated by the local search algorithm.
- T is a tree spanning all vertices for G, rooted at v^*.
- If we trace through T systematically (eg. by a DFS), we can enumerate all vertices.
- The major operation here is tracing each edge against its orientation (reversing the local search algorithm)
- No information regarding visited vertices needs to be stored since T is a tree.

Reverse Search for Enumeration

Algorithm 1: ReverseSearch(v)

```
output $v$
foreach neighbour $w$ of $v$ do
    if $f(w) = v$ then
        ReverseSearch($w$)
```

where f is the local search function.

To iterate over all vertices of G, we run ReverseSearch(v^*)

Applying Reverse Search to Pseudo Clique Enumeration

- We want to apply this idea to enumerate over all pseudo cliques of a given graph

- We need:
  - a way to score pseudo cliques
  - a definition of adjacent pseudo cliques
  - a parent-child relationship to define a traversal tree over all pseudo cliques

Pseudo Clique Enumeration: Basic Definitions

- Let $G = (V, E)$ be a graph with vertex set V and edge set E
- For a vertex set $U \subseteq V$, $E[U]$ is the set of edges whose endpoints are both in U
- $G[U] = (U, E[U])$ is the vertex induced subgraph by U
- the density of a vertex induced subgraph is defined as $G[U] = |E[U]|/clq(|U|)$, where $clq(n)$ is the number of edges in a clique of n vertices
- For a given threshold θ , $0 \leq \theta \leq 1$, $G[U]$ is a considered a pseudo clique if the density of $G[U]$ is no less than θ

Pseudo Clique Enumeration: Defining a Parent

Lemma 1: Let v be a vertex in $G[K]$ with the degree no greater than the average degree in $G[K]$. The density of $K - \{v\}$ is no less than the density of K.

- For any pseudo clique K, $K - \{v\}$ is also a pseudo clique.
- Since any K will always have such a vertex, vertices can be iteratively removed from K until $K = \emptyset$, passing through only pseudo cliques
- This definition of adjacency spans all pseudo cliques
- The graph induced by this adjacency is not a tree

Pseudo Clique Enumeration: Defining a Parent

- For a vertex set $K \neq \emptyset$, we define $v^*(K)$ to be the vertex with minimum degree in $G[K]$. If there are two vertices of minimum degree, take the lexicographically smaller one.
- Define the parent $prt(K)$ of K by $K - \{v^*(K)\}$
- If K is a pseudo clique, $prt(K)$ is a pseudo clique
- The graph induced by this parent-child relation forms a tree
- The definition of a parent does not depend on the threshold value, so the parent-child relationship is identical for all threshold values.

Pseudo Clique Enumeration: Defining Children

- The definition of the set of children of a given pseudo clique K is obtained directly from the definition of the parent.
- For a pseudo clique $K \subseteq V$, K' is a child of K if and only if $K' - K = \{v^*(K')\}$
- We can list the children of K by computing the density of $K \cup \{v\}$ and $\{v^*(K')\}$ for each vertex $v \notin K$
- K has at most $|V| - |K|$ children

Reverse Search for Pseudo Clique Enumeration

Algorithm 2: EnumeratePseudoCliques($G = (V, E), K$)

```
output $K$
foreach $v \notin K$ do
    if $K \cup \{v\}$ is a pseudo clique then
        if $v = \{v^*(K \cup \{v\})$ then
            EnumeratePseudoCliques($G = (V, E), K \cup \{v\}$)
```

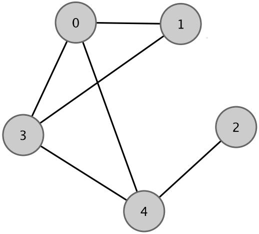
To iterate over all pseudo cliques of G, we run EnumeratePseudoCliques($G = (V, E), \{\}$)

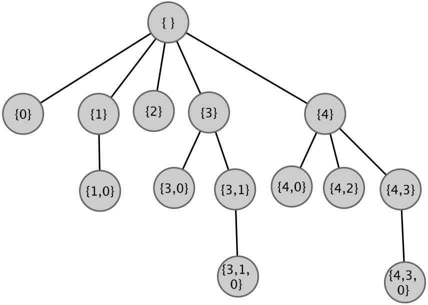
Reverse Search for Pseudo Clique Enumeration Example

$G = (V, E)$ where,

- $V = \{0, 1, 2, 3, 4\}$
- $E = \{\{0, 1\}, \{0, 3\}, \{0, 4\}, \{1, 3\}, \{2, 4\}, \{3, 4\}\}$

EnumeratePseudoCliques(G, { }) with $\theta = 1$





Identifying Novel Human Protein Complexes

- The human predicted protein-protein interaction graph has:
  - 172,184 interactions (edges), 130,470 which are novel predictions made by PIPE
  - up to 22,513 proteins (data set has not been completely compared)

- Next steps:
  - Run code on graph to identify all potential complexes with a relatively low θ
  - Filter list for complexes with
    - 4-12 proteins
    - a mix of known and predicted interactions

Conclusion

- Proteins, protein-protein interactions and protein complexes
- Reverse search for enumeration
- Pseudo clique enumeration using reverse search
- Plans to identify novel human protein complexes