CS4220: Knowledge Discovery Methods for Bioinformatics
Unit 6: Protein-Complex Prediction

Wong Limsoon
Lecture outline

• Overview of protein-complex prediction
• A case study: MCL-CAw
• Impact of PPIN cleansing
• Detecting overlapping complexes
• Detecting sparse complexes
• Detecting small complexes
Overview of Protein-Complex Detection from PPIN
“Assemblies” of interacting proteins

- Proteins interact to form “protein assemblies”
- These assemblies are like “protein machines”
  - Highly coordinated parts
  - Highly efficient

- Protein assemblies
  - Complexes
  - Functional modules
  - Intricate, ubiquitous, control many biological processes

Individual proteins come together and interact

Protein assembly of multiple proteins
Protein interaction networks

- Proteins come together & interact
- The collection of these interactions form a Protein Interaction Network or PPIN

Collection of such interactions in an organism

Valuable source of knowledge
Detection & analysis of protein complexes in PPIN

Identifying embedded complexes

Entire module might be involved in the same function/process

Individual complexes (Some might share proteins)

Space-time info is lost

Space-time info is “recovered”

Embedded complexes identified from PPIN

PPIN derived from several high-throughput expt
Identifying complexes from PPIN: The complete picture

1. Affinity purification followed by MS for identifying “baits” and “preys” (in vitro)
2. Arriving at a close approximation to the in vivo network
3. Identifying complexes from the PPI network
Taxonomy of Protein-complex prediction methods

Complex detection methods

Solely graph clustering
- Merging and growing clusters
  - MCODE (Bader et al., 2003)
  - LCMA (Li et al., 2005)
  - CMC (Liu et al., 2009)
  - HACO (Wang et al., 2009)
- Network partitioning
  - MCL (Dongen, 2000)
  - Pu et al. (2007)
  - Friedel et al. (2008)

Graph clustering + Biological information
- Core-attachment
- Functional homogeneity
- Evolutionary conservation
- Co-operative and exclusive interactions
  - Ozawa et al. (2010)
  - Jung et al. (2010)

Merging and growing clusters
- COACH (Wu et al., 2009)
- CORE (Leung et al., 2009)
- HUNTER (Chin et al., 2010

Merging and growing clusters
- PCP (Chua et al., 2007)
- DECAFF (Li et al., 2007)

Network partitioning
- RNSC (King et al., 2005)

Network alignment
- Sharan et al. (2004)
- Sharan et al. (2006)
- QNet (2007)
• As researchers try to improve basic graph clustering techs, they also incorporate bio insights into the methods.
Graph clustering: MCODE

- Good visualization
  - MCODE offered as a “plug-in” to Cytoscape
- Produces very few clusters
  - High accuracy, but low recall
- Performs well on highly filtered high-density PPIN
  - Low tolerance to noise

Seed a complex and look in neighborhood

- Weight vertices by density of their immediate neighbourhood
- Select vertices in decreasing order of weights
- ‘Seed’ a complex using vertex s
- Look in neighborhood of s
  - Vertex Weight Parameter
  - Add vertices to “grow” the complex

Graph clustering: MCL

- Popular software for general graph clustering
- Reasonably good for protein complex detection
- Highly scalable and fast; robust to noise

Repeated inflation and expansion separates the network into multiple dense regions

\[
(\Gamma_k M)_{pq} = \frac{(M_{pq})^r}{\sum_{i=1}^{k} (M_{iq})^r}
\]


Markov Chains

- To see how this works, an example:

  ![Graph](http://www.cs.ucsb.edu/~xyan/classes/CS595D-2009winter/MCL_Presentation2.pdf)

- In one time step, a random walker at node 1 has a 33% chance of going to node 2, 3, & 4, and 0% chance to nodes 5, 6, or 7.
- From node 2, 25% chance for 1, 3, 4, 5 and 0% for 6 and 7.
- Creating a transition matrix gives:

  
  \[
  \begin{pmatrix}
  0 & 0.25 & 0.33 & 0.33 & 0 & 0 & 0 \\
  0.33 & 0 & 0.33 & 0.33 & 0.33 & 0 & 0 \\
  0.33 & 0.25 & 0 & 0.33 & 0 & 0 & 0 \\
  0.33 & 0.25 & 0.33 & 0 & 0 & 0 & 0 \\
  0 & 0.25 & 0 & 0 & 0 & 0.5 & 0.5 \\
  0 & 0 & 0 & 0 & 0.33 & 0 & 0.5 \\
  0 & 0 & 0 & 0 & 0.33 & 0.5 & 0
  \end{pmatrix}
  \]

  (notice each column sums to one)

  Also can be looked at as a probability matrix!
Markov Chains

- A simpler example: \[
\begin{pmatrix}
.6 & .2 \\
.4 & .8 \\
\end{pmatrix}
\]

- Next time step: \[t_0 \rightarrow t_1 \rightarrow t_2\]

\[.6 \rightarrow .1 \rightarrow .1 + .1 \rightarrow .2 \rightarrow .1\]

\[.6 \times .6 + .4 \times .2 = .44\]

\[
\begin{pmatrix}
.6 & .2 \\
.4 & .8 \\
\end{pmatrix}
\begin{pmatrix}
.6 & .2 \\
.4 & .8 \\
\end{pmatrix}
= \begin{pmatrix}
.44 & .28 \\
.56 & .72 \\
\end{pmatrix}
\rightarrow \begin{pmatrix}
.35 & .32 \\
.65 & .68 \\
\end{pmatrix}
\rightarrow \begin{pmatrix}
.34 & .33 \\
.66 & .66 \\
\end{pmatrix}
\]

Eventually

\[
\begin{pmatrix}
.33 & .33 \\
.66 & .66 \\
\end{pmatrix}
\]
MCL

- "Flow is easier within dense regions than across sparse boundaries, however, in the long run this effect disappears."
- During the earlier powers of the Markov Chain, the edge weights will be higher in links that are within clusters, and lower between the clusters.
- This means there is a correspondence between the distribution of weight over the columns and the clusterings.
MCL

- MCL deliberately boosts this affect by
  - Stopping partway in the Markov Chain
  - Then adjusting the transitions by columns.
    For each vertex, the transition values are changed so that
    • Strong neighbors are further strengthened
    • Less popular neighbors are demoted.

- This adjusting can be done by raising a single column
to a non-negative power, and then re-normalizing.

- This operation is named “Inflation”

- (Taking the Markov Chain powers is named “Expansion”)
Evolutionary insight: Conserved subnets

**Assumption**
- Complexes are evolutionarily conserved

- Form orthology network out of PPINs from multiple species
- Identify conserved subnetworks
- Verify if these are complexes

Functional info: RNSC & DECAFF

- **RNSC**
  - Iterative clustering based on optimizing a cost function
  - Post-process based on size, edge-density, & functional homogeneity

- **DECAFF**
  - Li et al. *CSB 2007*, pp. 157-168
  - Find dense local neighborhoods and identify local cliques
  - Merge cliques to produce candidate complexes
  - Post-process based on functional homogeneity
Core-attachment structure: COACH

- Perform well on high-density PPIN
  - Higher recall than MCODE & MCL
- List cores & attachments separately

Mutually exclusive PPIs: SPIN

- +15% in precision & +10% in recall for MCL & MCODE using SPIN
- Limitation: Insufficient amt of domain-domain interaction data

Jung et al., Bioinformatics, 26(3):385-391, 2010

Fig. 6. Comparisons among the known complexes and clusters predicted by LCMA based on PPIN and SPIN. The gray ovals represent known complexes from MIPS, the quadrangle is a PPIN cluster, and the dotted quadrangles are SPIN clusters. A protein that appears in several complexes is underlined.
Statistics of yeast complexes

(a) Size distribution

(b) Large complexes

Figure 2.4: Statistics of the yeast reference complexes, from the CYC2008 database. (a) The size distribution of the complexes. (b) EXT (number of highly-connected external proteins) and DENS (density) distributions of large complexes.
Figure 2.6: Performance of the ten clustering algorithms on prediction of yeast complexes, with (a) $match\_thresh = 0.75$ for large complexes, (b) $match\_thresh = 1$ for small complexes. The left chart shows the precision, recall, F score, and AUC of the precision-recall graph. The right chart shows the precision-recall graph.
What they do badly on Low DENS, high EXT complexes are difficult. Small complexes are difficult. High DENS, low EXT complexes are predicted at higher score. Low DENS, high EXT complexes are predicted with extra proteins. High EXT complexes are predicted as merged clusters.

Figure 2.8: Performance of complex-discovery algorithms on yeast complexes, stratified by size, DENS, and EXT. The x-axis of each chart corresponds to the different stratified groups of complexes, given at the bottom of the figure.
Challenges

• Recall & precision of protein complex prediction algo’s have lots to be improved
  – Does a “cleaner” PPIN help?

• How to capture “high edge density” complexes that overlap each other?

• How to capture “low edge density” complexes?

• How to capture small complexes?
A Case Study: MCL-CAw
Core-attachment modularity in yeast complexes

- **Cores**
  - High interactivity among each other
  - Highly co-expressed
  - Main functional units of complexes

- **Attachments**
  - Not co-expressed w/ cores all the time
  - Attach to cores & aid them in their functions
  - May be shared across complexes

MCL-CAw: Key idea

- Identify **dense** regions within PPI network
- Identify **core-attachment** structures within these regions
- Extract them out, discard the rest

MCL-CAw: Main steps

- Cluster PPI network using MCL hierarchically
- Identify core proteins within clusters
- Filter noisy clusters
- Recruit attachment proteins to cores
- Extract out complexes
- Rank the complexes
Step 1: Cluster by MCL hierarchically

Why MCL?
- Simple, robust, scalable
- Find dense regions reasonably well
- Work on weighted networks

Why hierarchical?
- Some clusters are large, & amalgamate smaller ones
- Hierarchical clustering identifies these smaller clusters

PPI Network

MCL clustering

Amalgamated cluster

Hierarchical MCL clustering of large clusters

Apply MCL to large clusters to break them up
Step 2: Identify core proteins in clusters

- Set of cores within a cluster:
  - Essentially a k-core
  - But, with some additional restrictions

Protein $p \in \text{Core } (C_i)$ if:
1. $p$ has high degree w.r.t. $C_i$
2. $p$ has more neighbors within $C_i$ than outside

Protein $p \in \text{Core } (C_i)$ if:
1. $\text{In-degree of } p \text{ w.r.t. } C_i \geq \text{Avg in-degree of } C_i$
2. $\text{In-degree of } p \text{ w.r.t. } C_i > \text{Out-degree of } p \text{ w.r.t. } C_i$
   (Considering weighted degrees)
Step 3: Filter noisy clusters

- In accordance with our assumption that every complex we predict must have a core

- Discard noisy clusters (i.e., those w/o core)
Step 4: Identify attachments to cores

- Protein \( p \) is an attachment to an acceptor cluster, if
  1. Non-core
  2. Has strong interactions with core proteins
  3. Stronger the interactions among cores, stronger have to be the interactions of \( p \)
  4. Large core sets, strong interactions to some, or weaker to many

\[
\text{Interactions} \left( p, \text{Core}(C_j) \right) \propto \text{Interactions}(\text{Core}(C_j))
\]
Step 4: Identify attachments to cores

**Protein** $p \in \text{Donor cluster } C_i$ is an attachment to Accoptor Core ($C_j$), if:

$$l(p, \text{Core}(C_j)) \geq \alpha \times l(\text{Core}(C_j)) \times [\frac{|\text{Core}(C_j)|}{2}]^{-\gamma}$$

Parameters $\alpha$ and $\gamma$ used to control effect of right-hand side
Step 5: Extract complexes

Complex $C = \text{Core}(C) \cup \text{Attach}(C)$

- Attachment proteins may be shared betw complexes
Step 6: Rank predicted complexes

• **Weighted density-based ranking of complexes**
  – Reliability of interactions within complex C
  – Size of complex C

– Weighted density

\[ \text{Weighted density} = \frac{\sum (\text{wt of interactions})}{(|C| \times (|C|-1))} \]

• **Unweighted density** → Blindly favors small complexes or complexes with large # of interactions

• **Weighted density** → More reliable complexes ranked higher
PPI datasets for evaluation of MCL-CAw

- **Unscored,**
  - G+K: Gavin and Krogan datasets combined

- **Scored**
  - G+K (ICD): Scoring G+K network by iterated CD distance
  - A few other edge weighting schemes are also used

If you don’t remember CD-distance, please refer to last lecture!
“Gold standard” benchmarks complexes

- **CYC 08: 408 complexes**

- **MIPS: 313 complexes,**

- **Aloy: 101 complexes,**

<table>
<thead>
<tr>
<th>Datasets</th>
<th>#cmplx</th>
<th>#proteins</th>
<th>size</th>
<th>density</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>max</td>
<td>avg</td>
</tr>
<tr>
<td>Aloy</td>
<td>63</td>
<td>544</td>
<td>34</td>
<td>9.22</td>
</tr>
<tr>
<td>CYC08</td>
<td>148</td>
<td>1115</td>
<td>81</td>
<td>8.84</td>
</tr>
<tr>
<td>MIPS</td>
<td>156</td>
<td>1171</td>
<td>95</td>
<td>14.86</td>
</tr>
<tr>
<td>Combined</td>
<td>305</td>
<td>1543</td>
<td>95</td>
<td>11.85</td>
</tr>
</tbody>
</table>

Size > 3: Measured based on BioGrid yeast physical PPIN
Evaluation of MCL-CAw

<table>
<thead>
<tr>
<th>Method</th>
<th>F1</th>
<th>Norm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. CMC</td>
<td>1.146</td>
<td>1.000</td>
</tr>
<tr>
<td>2. HACO</td>
<td>0.899</td>
<td>0.785</td>
</tr>
<tr>
<td>3. MCL-CAw</td>
<td>0.800</td>
<td>0.700</td>
</tr>
<tr>
<td>4. CORE</td>
<td>0.757</td>
<td>0.661</td>
</tr>
<tr>
<td>5. MCLO</td>
<td>0.734</td>
<td>0.641</td>
</tr>
<tr>
<td>6. MCL</td>
<td>0.717</td>
<td>0.626</td>
</tr>
<tr>
<td>7. COACH</td>
<td>0.515</td>
<td>0.450</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>F1</th>
<th>Norm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. MCL-CAw</td>
<td>1.595</td>
<td>1.000</td>
</tr>
<tr>
<td>2. HACO</td>
<td>1.536</td>
<td>0.962</td>
</tr>
<tr>
<td>3. CMC</td>
<td>1.516</td>
<td>0.950</td>
</tr>
<tr>
<td>4. MCLO</td>
<td>1.414</td>
<td>0.886</td>
</tr>
<tr>
<td>5. MCL</td>
<td>1.411</td>
<td>0.884</td>
</tr>
</tbody>
</table>

Adding the F1 scores across all three benchmarks and normalizing against the best.

- CORE and COACH assume only unweighted networks

- F1 values have increased for all methods upon scoring
Strengths of MCL-CAw

• **Perform better than MCL**
  – Demonstrate effectiveness of adding biological insights (core-attachment structure)

• **Respond well to most affinity-scoring schemes**
  – Always ranked among top 3 on all scored / weighted networks
  – Weighting of edges improves performance of MCL-Caw and other methods
    • Good to incorporate reliability info of the edges!
Limitations of MCL-CAw

• **Amalgamation of closely-interacting complexes**
  – “Inherited” from MCL
  – Lowers the recall

• **Undetected sparse complexes**
  – “Inherited” from MCL
  – Does not work when PPI is sparse
  – Less sensitive to very sparse complexes

• **Undetected small complexes (size < 4)**
  – Discards small predicted complexes as many are FP
Impact of PPIN Cleansing on Protein Complex Prediction
Cleaning PPIN

• Modify existing PPI network as follow
  – Remove interactions with low weight
  – Add interactions with high weight

• Then run RNSC, MCODE, MCL, …, as well as our own method CMC
CMC: Clustering of Maximal Cliques

- Remove noise edges in input PPI network by discarding edges having low iterated CD-distance

- Augment input PPI network by addition of missing edges having high iterated CD-distance

- Predict protein complex by finding overlapping maximal cliques, and merging/removing them

- Score predicted complexes using cluster density weighted by iterated CD-distance

Some details of CMC

• Iterated CD-distance is used to weigh PPI’s

\[ w^k(u, v) = \frac{\sum_{x \in N_u \cap N_v} (w^{k-1}(x, u) + w^{k-1}(x, v))}{\sum_{x \in N_u} w^{k-1}(x, u) + \lambda_u^k + \sum_{x \in N_v} w^{k-1}(x, v) + \lambda_v^k} \]

• Clusters are ranked by weighted density

\[ \text{score}(C) = \frac{\sum_{u \in C, v \in C} w(u, v)}{|C| \cdot (|C| - 1)} \]

• Inter-cluster connectivity is used to decide whether highly overlapping clusters are merged or (the lower weighted density ones) removed

\[ \text{inter-score}(C_1, C_2) = \sqrt{\frac{\sum_{u \in (C_1 - C_2)} \sum_{v \in C_2} w(u, v)}{|C_1 - C_2| \cdot |C_2|} \cdot \frac{\sum_{u \in (C_2 - C_1)} \sum_{v \in C_1} w(u, v)}{|C_2 - C_1| \cdot |C_1|}} \]
Validation experiments

• **Matching a predicted complex S with a true complex C**
  - Vs: set of proteins in S
  - Vc: set of proteins in C
  - \( \text{Overlap}(S, C) = \frac{|Vs \cap Vc|}{|Vs \cup Vc|} \), \( \text{Overlap}(S, C) \geq 0.5 \)

• **Evaluation**
  - Precision = matched predictions / total predictions
  - Recall = matched complexes / total complexes

• **Datasets: combined info from 6 yeast PPI expts**
  - #interactions: 20,461 PPI from 4,671 proteins
  - #interactions with >0 common neighbor: 11,487
Effecting of cleaning on CMC

- Cleaning by Iterated CD-distance improves recall & precision of CMC
Noise tolerance of CMC

- If cleaning is done by iterating CD-distance 20 times, CMC can tolerate up to 500% noise in the PPI network!
Effect of cleansing on MCL

- MCL benefits significantly from cleaning too
Table 3. The impact of the iterative scoring method on the performance of four clustering methods. For CMC, MCL and CFinder, we retain only the top-6000 interactions, and no new interactions are added. For MCode, we retain all the interactions with non-zero score and add top-3000 new interactions with the highest score. The 2nd column is the number of iterations $k$ of the iterative scoring method, and $k=0$ means the PPI network is unweighted. The 3rd column is the number of clusters generated, the 4th and 5th column is the average size and co-localization score of generated clusters.
Characteristics of unmatched clusters

- At $k = 2$ …
- 85 clusters predicted by CMC do not match complexes in Aloy and MIPS
- Localization coherence score ~90%
- 65/85 have the same informative GO term annotated to > 50% of proteins in the cluster

$\Rightarrow$ Likely to be real complexes
Detecting Overlapping Protein Complexes from Dense Regions of PPIN
Overlapping complexes in dense regions of PPIN

- Dense regions of PPIN often contain multiple overlapping protein complexes
- These complexes often got clustered together and cannot be corrected detected

Two ideas to cleanse PPI network
- Decompose PPI network by localisation GO terms
- Remove big hubs

Idea I: Split by localization GO terms

• A protein complex can only be formed if its proteins are localized in the same compartment of the cell

⇒ Use **general** cellular component (CC) GO terms to decompose a given PPI network into several smaller PPI networks

• Use “general” CC GO terms as it is easier to obtain rough localization annotation of proteins
  – How to choose threshold \( N_{GO} \) to decide whether a CC GO term is “general”? 
Effect of $N_{GO}$ on precision

- Precision always improves under all $N_{GO}$ thresholds
Effect of $N_{GO}$ on recall

- Recall drops when $N_{GO}$ is small due to excessive info loss

- Recall improves when $N_{GO} > 300$  
  $\Rightarrow$ Good to decompose by general CC GO terms
Idea II: Remove big hubs

- Hub proteins are those proteins that have many neighbors in the PPI network

- Large hubs are likely to be “date hubs”; i.e., proteins that participate in many complexes
  - Likely to confuse protein complex prediction algo

⇒ Remove large hubs before protein complex prediction
  - How to choose threshold \( N_{hub} \) to decide whether a hub is “large”?
Effect of $N_{\text{hub}}$ on recall

- Recall is affected when $N_{\text{hub}}$ is small, due to high info loss
- Not much effect on recall when $N_{\text{hub}}$ is large
Effect of $N_{\text{hub}}$ on precision

- Precision of MCL & RNSC not much change
- Precision of IPCA & CMC improve greatly

Table 5. Localization coherence score of generated clusters when different $N_{\text{hub}}$ values are used for removing hub proteins.
Combining the two ideas

1. Let $\mathcal{C}$ be the set of clusters generated. Initially $\mathcal{C}$ is empty.

2. Remove hub proteins that have at least $N_{\text{hub}}$ neighbors from the given PPI network $G$. Let $G'$ be the resultant network.

3. Let $g_1, \ldots, g_m$ be the localization GO terms that are selected using threshold $N_{\text{GO}}$. For each $g_i$, do the following:
   
   • Remove proteins that are not annotated with $g_i$ from $G'$. Let $G'_i$ be the resultant network.
   
   • Apply a complex discovery algorithm on $G'_i$ to find clusters. Let $\mathcal{C}_i$ be the set of clusters generated.

   • $\mathcal{C} = \mathcal{C} \cup \mathcal{C}_i$;

4. Remove duplicated clusters from $\mathcal{C}$. 
Effect of combining $N_{GO}$ & $N_{hub}$

- RNSC doesn’t benefit further
- MCL, IPCA & CMC all gain further

### Table 5 - F1-measure of the four algorithms when $match\_thres=0.5$

<table>
<thead>
<tr>
<th></th>
<th>original</th>
<th>Hub50</th>
<th>GO500</th>
<th>Hub50+GO500</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCL</td>
<td>0.250</td>
<td>0.272</td>
<td>0.354</td>
<td>0.406</td>
</tr>
<tr>
<td>RNSC</td>
<td>0.353</td>
<td>0.347</td>
<td>0.471</td>
<td>0.436</td>
</tr>
<tr>
<td>IPCA</td>
<td>0.191</td>
<td>0.405</td>
<td>0.368</td>
<td>0.469</td>
</tr>
<tr>
<td>CMC</td>
<td>0.207</td>
<td>0.421</td>
<td>0.359</td>
<td>0.501</td>
</tr>
</tbody>
</table>
Conclusions

- RNSC performs best ($F1 = 0.353$) on original PPI network; it also benefits much from CC GO term decomposition, but not from big-hub removal.

- CMC performs best ($F1 = 0.501$) after PPI network preprocessing by CC GO term decomposition and big-hub removal.

- But many complexes still cannot be detected…
Time for Exercise #1

- Removal of some big hubs is not necessarily beneficial to protein-complex prediction from PPIN. Discuss how one can go about identifying those big hubs whose removal are beneficial.
Why many complexes are not detectable

- Among 305 complexes, 81 have density < 0.5, and 42 have density < 0.25
Why many complexes are not detectable

- 18 complexes w/ more than half of their proteins being isolated
  - Isolated vertex connects to no other vertices in the complex

- 144 complexes w/ more than half of their proteins being loose
  - Loose vertex connects to < 50% of other vertices in the complex
Why many complexes are not detectable

- For all four algo’s, 90% of detected complexes have a density > 0.5
- But many undetected complexes have a density < 0.5, and also have many loose vertices
Detecting Protein Complexes from Sparse Regions of PPIN
ANY algorithm based solely on topological will miss these sparse complexes!!
Noisy & transient PPIs

• **Noise in PPI data**
  – Spuriously-detected interactions (false positives), and missing interactions (false negatives)

• **Transient interactions**
  – Many proteins that actually interact are not from the same complex, they bind temporarily to perform a function

• **Also, not all proteins in the same complex may actually interact with each other**
Cytochrome BC1 complex

- Involved in electron-transport chain in mitochondrial inner membrane

- Discovery of this complex from PPI data is difficult
  - Sparseness of the complex’s PPI subnetwork
    - Only 19 out of 45 possible interactions were detected between the complex’s proteins
  - Many extraneous interactions detected with other proteins outside the complex
    - E.g., UBI4 is involved in protein ubiquitination, and binds to many proteins to perform its function.
• Key idea to deal with sparseness

Augment physical PPI network with other forms of linkage that suggest two proteins are likely to interact

Supervised Weighting of Composite Networks (SWC)

• Data integration
• Supervised edge weighting
• Clustering

Overview of SWC

1. Integrate diff data sources to form composite network

2. Weight each edge based on probability that its two proteins are co-complex, using a naïve Bayes model w/ supervised learning

3. Perform clustering on the weighted network

• Advantages
  – Data integration increases density of complexes
    • co-complex proteins are likely to be related in other ways even if they do not interact
  – Supervised learning
    • Allows discrimination betw co-complex and transient interactions
  – Naïve Bayes’ transparency
    • Model parameters can be analyzed, e.g., to visualize the contribution of diff evidences in a predicted complex
1. Integrate multiple sources

- **Composite network**: Vertices represent proteins, edges represent relationships between proteins.
- **There is an edge betw proteins** u, v, *if and only if* u and v are related according to any of the data sources.

<table>
<thead>
<tr>
<th>Data source</th>
<th>Database</th>
<th>Scoring method</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPI</td>
<td>BioGRID, IntACT, MINT</td>
<td>Iterative AdjustCD.</td>
</tr>
<tr>
<td>L2-PPI (indirect PPI)</td>
<td>BioGRID, IntACT, MINT</td>
<td>Iterative AdjustCD</td>
</tr>
<tr>
<td>Functional association</td>
<td>STRING</td>
<td>STRING</td>
</tr>
<tr>
<td>Literature co-occurrence</td>
<td>PubMed</td>
<td>Jaccard coefficient</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data source</th>
<th>Yeast</th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># Pairs</td>
<td>% co-complex</td>
</tr>
<tr>
<td>PPI</td>
<td>106328</td>
<td>5.8%</td>
</tr>
<tr>
<td>L2-PPI</td>
<td>181175</td>
<td>1.1%</td>
</tr>
<tr>
<td>STRING</td>
<td>175712</td>
<td>5.7%</td>
</tr>
<tr>
<td>PubMed</td>
<td>161213</td>
<td>4.9%</td>
</tr>
<tr>
<td>All</td>
<td>531800</td>
<td>2.1%</td>
</tr>
</tbody>
</table>
2. Supervised edge-weighting

- Treat each edge as an instance, where features are data sources and feature values are data source scores, and class label is “co-complex” or “non-co-complex”

<table>
<thead>
<tr>
<th>PPI</th>
<th>L2 PPI</th>
<th>STRING</th>
<th>Pubmed</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.56</td>
<td>451</td>
<td>0</td>
<td>“co-complex”</td>
</tr>
<tr>
<td>0.1</td>
<td>0</td>
<td>25</td>
<td>0</td>
<td>“non-co-complex”</td>
</tr>
</tbody>
</table>

- Supervised learning:
  1. Discretize each feature (Minimum Description Length discretization\(^7\))
  2. Learn maximum-likelihood parameters for the two classes:

\[
P(F = f|\text{co - comp}) = \frac{n_{c,F=f}}{n_c} \quad P(F = f|\text{non - co - comp}) = \frac{n_{-c,F=f}}{n_{-c}}
\]

for each discretized feature value \( f \) of each feature \( F \)

- Weight each edge \( e \) with its posterior probability of being co-complex:

\[
\text{weight}(e) = \frac{P(F = f|\text{co - comp}) P(F_1 = f_1, F_2 = f_2, \ldots) Z}{P(F_1 = f_1, F_2 = f_2, \ldots | \text{co - comp}) P(\text{co - comp})} = \frac{\prod_i P(F_i = f_i|\text{co - comp}) P(\text{co - comp}) Z}{\prod_i P(F_i = f_i|\text{co - comp}) P(\text{co - comp}) + \prod_i P(F_i = f_i|\text{non - co - comp}) P(\text{non - co - comp})}
\]
3. Complex discovery

- Weighted composite network used as input to clustering algorithms
  - CMC, ClusterONE, IPCA, MCL, RNSC, HACO
- Predicted complexes scored by weighted density

- The clustering algo’s generate clusters with low overlap
  - Only 15% of clusters are generated by two or more algo’s
  ⇒ Voting-based aggregative strategy, COMBINED:
    - Take union of clusters generated by the diff algo’s
    - Similar clusters from multiple algo’s are given higher scores
      - If two or more clusters are similar (Jaccard >= 0.75), then use the highest scoring one and multiply its score by the # of algo’s that generated it
Experiments

• Weighting approaches:
  – SWC vs BOOST, TOPO, STR, NOWEI

• Evaluate performance on the 6 clustering algos and the COMBINED clustering strategy

• Real complexes for training and testing: CYC200814 for yeast, CORUM15 for human

• Evaluation
  – How well co-complex edges are predicted
  – How well predicted complexes match real complexes
Evaluation wrt co-complex prediction

Figure 2 Precision-recall graph for classification of co-complex edges using the five weighting schemes. (a) Classification of yeast co-complex edges. SWC and BOOST achieve the highest recall through data integration. TOPO has high precision for its top-scoring edges, but these are clustered in a few complexes. SWC achieves higher precision than STR, except when too many edges are considered. BOOST classifies edges categorically, giving high scores to one set of edges with about 50% recall and 35% precision, and low scores to the remainder. (b) Classification of human co-complex edges. Recall and precision for human is much lower than for yeast. TOPO has higher precision than SWC, but its predicted edges are clustered in fewer complexes. BOOST classifies edges categorically, and its high-scoring edges achieve 7% recall, with comparable precision with SWC. NOWEI has slightly higher precision than STR, which has the lowest precision.
Evaluation wrt yeast complex prediction
Evaluation wrt human complex prediction
Yeast BC1 complex

PPI network

Composite network

SWC-weighted network

Likelihood network

- protein from complex
- protein outside complex
- PPI
- STRING
- PUBMED
- SWC weighted edge
- predicted cluster
Human BRCA1-A complex

- SWC found a complex that included 5 extra proteins, of which 3 (BABAM1, BRE, BRCC3) have been included in the BRCA1-A complex.
High-confidence predicted complexes

Yeast

Human

# of predictions

Biological process coherence

# of predictions

Biological process coherence
Two novel predicted complexes

- **Novel yeast complex**: Annotated with DNA metabolic process and response to stress, forms a complex called Cul8-RING which is absent in our ref set.

- **Novel human complex**: Annotated with transport process, Uniprot suggests it may be a subunit of a potassium channel complex.
## Novel complexes predicted

### Yeast

<table>
<thead>
<tr>
<th>Biological process</th>
<th># complexes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protein metabolic process</td>
<td>49</td>
</tr>
<tr>
<td>RNA metabolic process</td>
<td>36</td>
</tr>
<tr>
<td>DNA metabolic process</td>
<td>15</td>
</tr>
<tr>
<td>Small molecule metabolic process</td>
<td>23</td>
</tr>
<tr>
<td>Regulation of metabolic process</td>
<td>11</td>
</tr>
<tr>
<td>Regulation of gene expression</td>
<td>8</td>
</tr>
<tr>
<td>Organelle organization</td>
<td>40</td>
</tr>
<tr>
<td>Transport</td>
<td>43</td>
</tr>
<tr>
<td>Response to stress</td>
<td>20</td>
</tr>
<tr>
<td>Response to chemical stimulus</td>
<td>7</td>
</tr>
<tr>
<td>Cell cycle process</td>
<td>11</td>
</tr>
</tbody>
</table>

### Human

<table>
<thead>
<tr>
<th>Biological process</th>
<th># complexes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protein metabolic process</td>
<td>32</td>
</tr>
<tr>
<td>RNA metabolic process</td>
<td>29</td>
</tr>
<tr>
<td>DNA metabolic process</td>
<td>4</td>
</tr>
<tr>
<td>Small molecule metabolic process</td>
<td>19</td>
</tr>
<tr>
<td>Regulation of metabolic process</td>
<td>74</td>
</tr>
<tr>
<td>Regulation of gene expression</td>
<td>34</td>
</tr>
<tr>
<td>Organelle organization</td>
<td>19</td>
</tr>
<tr>
<td>Transport</td>
<td>38</td>
</tr>
<tr>
<td>Response to stress</td>
<td>28</td>
</tr>
<tr>
<td>Response to chemical stimulus</td>
<td>32</td>
</tr>
<tr>
<td>Cell cycle process</td>
<td>14</td>
</tr>
</tbody>
</table>
Conclusions

• **Naïve-Bayes data-integration to predict co-complexed proteins**
  
  – Use of multiple data sources increases density of complexes
  
  – Supervised learning allows discrimination between co-complex and transient interactions

• **Tested approach using 6 clustering algo’s**
  
  – Clusters produced by different algo’s have low overlap, combining them gives greater recall
  
  – Clusters produced by more algo’s are more reliable
Detecting Small Protein Complexes
Motivation

- Size of protein complexes follows a power-law distribution, meaning that most complexes are small (ie. 2 or 3 distinct proteins)
Time for Exercise #2

• Traditionally, complexes are predicted by searching for dense clusters in a PPI network. Discuss why such an approach cannot do well in predicting small protein complexes
Two-stage approach

1. Size-specific supervised weighting (SSS)

   - Discretize initial 12 features
   - Learn likelihood parameters for initial 12 features
   - Calculate posterior probabilities using initial 12 features
     
   - Derive ISO feature, discretize it
   - Learn likelihood parameters for ISO feature
   - Re-calculate posterior probabilities using all 13 features

2. Extract

   - Disambiguate posterior probabilities into size-2, size-3 components
   - Score each edge and triangle

---

Stage 1: SSS

1. Size-specific supervised weighting (SSS)
   - Discretize initial 12 features
   - Learn likelihood parameters for initial 12 features
   - Calculate posterior probabilities using initial 12 features
   - Derive ISO feature, discretize it
   - Re-calculate posterior probabilities using all 13 features
   - Learn likelihood parameters for ISO feature

2. Extract
   - Disambiguate posterior probabilities into size-2, size-3 components
   - Score each edge and triangle
Discretize initial 12 features

- Each edge in PPIN is cast as a data instance, with 12 initial features
  - 3 data sources
    - PPI (BioGrid + IntAct + MINT)
    - Functional associations (STRING)
    - Co-occurrence in literature (PUBMED)
  - 3 topological characteristics for each data source
    - Degree
    - Neighbourhood connectivity
    - Shared neighbours

- Discretize based on Minimum Description Length (MDL)
Stage 1: SSS

1. Size-specific supervised weighting (SSS)

   - Discretize initial 12 features
   - Learn likelihood parameters for initial 12 features
   - Calculate posterior probabilities using initial 12 features
   - Re-calculate posterior probabilities using all 13 features
   - Learn likelihood parameters for ISO feature
   - Derive ISO feature, discretize it

2. Extract

   - Disambiguate posterior probabilities into size-2, size-3 components
   - Score each edge and triangle
Learn likelihood parameters for initial 12 features

- Likelihood models for 3 classes (small co-complex, large co-complex, non co-complex)

\[
P(F = f | sm\text{-}comp) = \frac{n_{sm,F=f}}{n_{sm}}
\]

\[
P(F = f | lg\text{-}comp) = \frac{n_{lg,F=f}}{n_{lg}}
\]

\[
P(F = f | non\text{-}comp) = \frac{n_{non,F=f}}{n_{non}}
\]
Calculate posterior probabilities using initial 12 features

- Weight each edge with its posterior probability of being small co-complex, large co-complex, or non co-complex, using the naïve-Bayes formulation
  - Eg., probability that edge \((a, b)\) is small co-complex

\[
P((a, b) \text{ is } \text{sm-comp}|F_1 = f_1, F_2 = f_2, \ldots) = \frac{\prod_i P(F_i = f_i|(a, b) \text{ is } \text{sm-comp})P(\text{sm-comp})}{\sum_{\text{class} \in \{\text{sm-comp, lg-comp, non-comp}\}} \prod_i P(F_i = f_i|(a, b) \text{ is class})P(\text{class})}
\]

- These three probabilities are abbreviated as
  - \(P_{(a,b),\text{sm}}\)
  - \(P_{(a,b),\text{lg}}\)
  - \(P_{(a,b),\text{non}}\)
Stage 1: SSS

1. Size-specific supervised weighting (SSS)

   - Discretize initial 12 features
   - Learn likelihood parameters for initial 12 features
   - Calculate posterior probabilities using initial 12 features

   - Re-calculate posterior probabilities using all 13 features
   - Learn likelihood parameters for ISO feature

   - Derive ISO feature, discretize it

2. Extract

   - Disambiguate posterior probabilities into size-2, size-3 components
   - Score each edge and triangle
Derive ISO feature

• For each edge, derive a new feature, Isolatedness
  – Prob that the edge is isolated, or is part of an isolated triangle
  – Uses posterior prob calculated previously

\[
ISO(a, b) = ISO2(a, b) + ISO3(a, b)
\]

\[
ISO2(a, b) = P_{(a,b),sm} \prod_{x \in \{a,b\}, y \in N_{a,b}} P_{(x,y),non}
\]

\[
ISO3(a, b) = \sum_{c \in N_a \cap N_b} \left( P_{(a,b),sm} P_{(a,c),sm} P_{(b,c),sm} \prod_{x \in \{a,b,c\}, y \in N_{a,b,c}} P_{(x,y),non} \right)
\]

• This feature is also discretized using MDL
Stage 1: SSS

1. Size-specific supervised weighting (SSS)

- Discretize initial 12 features
- Learn likelihood parameters for initial 12 features
- Calculate posterior probabilities using initial 12 features
- Re-calculate posterior probabilities using all 13 features
- Learn likelihood parameters for ISO feature
- Derive ISO feature, discretize it

2. Extract

- Disambiguate posterior probabilities into size-2, size-3 components
- Score each edge and triangle
Learn likelihood parameters for ISO feature & Recalculate posterior prob using all 13 features

• Likelihood parameters are learned for the ISO feature in the same way as with the previous features

• Posterior prob are re-calculated as before, this time incorporating the new ISO feature
  – $P(a,b), sm = \text{prob that } (a,b) \text{ is small co-complex}$
  – $P(a,b), lg = \text{prob that } (a,b) \text{ is large co-complex}$
  – $P(a,b), non = \text{prob that } (a,b) \text{ is non co-complex}$
Stage 2: Extract

1. Size-specific supervised weighting (SSS)

   - Discretize initial 12 features
   - Learn likelihood parameters for initial 12 features
   - Calculate posterior probabilities using initial 12 features

   - Re-calculate posterior probabilities using all 13 features
   - Learn likelihood parameters for ISO feature
   - Derive ISO feature, discretize it

2. Extract

   - Disambiguate posterior probabilities into size-2, size-3 components
   - Score each edge and triangle
Disambiguate $P_{(a,b),sm}$, the prob that (a,b) is small co-complex, into size-2 and size-3 components

- If (a,b) is part of a high-weighted triangle, then it is likelier to be part of a size-3 complex, so reduce its size-2 component

\[
P'_{(a,b),sm2} = P_{(a,b),sm} - \sum_{x \in N_a \cap N_b} P_{(a,b),sm} P_{(a,x),sm} P_{(b,x),sm}
\]

(a,b) likelier to be part of a size-3 complex $abc$ than a size-2 complex $ab$

(a,b) likelier to be a size-2 complex than size-3 complex $abc$
Disambiguate $P_{(a,b), sm}$, the prob that $(a,b)$ is small co-complex, into size-2 and size-3 components

- If $(a,b)$ is part of a high-weighted triangle, and is part of another low-weighted triangle, then it is likelier to be in a complex with the first triangle

$$P'_{(a,b), sm3, abc} = P_{(a,b), sm} - \sum_{x \in N_a \cap N_b \setminus \{c\}} P_{(a,b), sm} P_{(a,x), sm} P_{(b,x), sm}$$

$(a,b)$ likelier to be part of a size-3 complex $abc$, than complex $abd$
Stage 2: Extract

1. Size-specific supervised weighting (SSS)

   - Discretize initial 12 features
   - Learn likelihood parameters for initial 12 features
   - Calculate posterior probabilities using initial 12 features
   - Re-calculate posterior probabilities using all 13 features
   - Learn likelihood parameters for ISO feature
   - Derive ISO feature, discretize it

2. Extract

   - Disambiguate posterior probabilities into size-2, size-3 components
   - Score each edge and triangle
Score each edge and triangle

• Every edge / triangle is taken as candidate size-2 / -3 complexes

• Score each candidate complex, using edges inside the complex, as well as outgoing edges from the complex
  – For each candidate complex, its score is its cohesiveness multiplied by its weighted density

• Cohesiveness:

\[
\frac{\sum \text{edge weights inside cluster}}{\sum \text{edge weights inside cluster} + \sum \text{outgoing edge weights from cluster}}
\]
The cohesiveness of a size-2 cluster \((a, b)\) and a size-3 cluster \((a, b, c)\) respectively are:

\[
Coh(a, b) = \frac{P'_{(a,b), sm2}}{P'_{(a,b), sm2} + \sum_{x \in \{a,b\}, y \in \text{Na}, b} (P_{(x,y), sm} + P_{(x,y), lg})}
\]

\[
Coh(a, b, c) = \frac{P'_{(a,b), sm3, abc} + P'_{(a,c), sm3, abc} + P'_{(b,c), sm3, abc}}{P'_{(a,b), sm3, abc} + P'_{(a,c), sm3, abc} + P'_{(b,c), sm3, abc} + \sum_{x \in \{a,b,c\}, y \in \text{Na}, b, c} (P_{(x,y), sm} + P_{(x,y), lg})}
\]

We then define the score of a cluster as its cohesiveness-weighted density, or the product of its weighted density and its cohesiveness. The score of a size-2 cluster \((a, b)\), and a size-3 cluster \((a, b, c)\) respectively are:

\[
\text{score}(a, b) = Coh(a, b)P'_{(a,b), sm2}
\]

\[
\text{score}(a, b, c) = Coh(a, b, c)\frac{(P'_{(a,b), sm3, abc} + P'_{(a,c), sm3, abc} + P'_{(b,c), sm3, abc})}{3}
\]
Two-stage approach

1. Size-specific supervised weighting (SSS)

Discretize initial 12 features

Learn likelihood parameters for initial 12 features

Calculate posterior probabilities using initial 12 features

Re-calculate posterior probabilities using all 13 features

Learn likelihood parameters for ISO feature

Derive ISO feature, discretize it

2. Extract

Disambiguate posterior probabilities into size-2, size-3 components

Score each edge and triangle
Benefits

• Groups of proteins may take on small-complex topological characteristics in PPIN by chance
  ⇒ Use multiple data sources & their topological features
  • Unlikely that all data sources share small-complex characteristics by chance

• Small-complex prediction is sensitive to noise in PPIN
  ⇒ Reduce noise by data integration with supervised learning

• Other supervised-weighting complex-prediction approaches learn features of large complexes
  – Do not perform well for small complexes
  ⇒ Size-specific weighting

• Scoring candidate small complexes is sensitive to correct edge weights (very few edge weights used for scoring)
  ⇒ Use also outgoing edges from candidate complex during scoring
Experiment

• Compare the following approaches:
  – SSS + Extract: Proposed approach
  – Standard algo’s with reliability-weighted PPI network (PPIREL)
  – Standard algo’s with SSS-weighted network

• 10 rounds of cross-validation

• Prediction of yeast small complexes, with CYC2008 yeast reference complexes (human complexes also evaluated in manuscript)

• Exact-match evaluation: Predicted complexes have to match reference complexes exactly
Yeast small-complex prediction
Human small-complex prediction

Not as good as for yeast. Why?

Figure 3 Performance of small complex prediction in human, (a) precision-recall AUC, (b) and (c) precision-recall graphs.
Quality of novel complexes predicted

(a) Yeast

(b) Human

Figure 10 Number of high-confidence novel predictions, and their semantic coherences, in (a) yeast, (b) human.
Yeast DNA replication factor A

(a) PPIREL network

(b) SSS network

- DNA replication factor A consists of 3 proteins
- Cannot be found by standard clustering algorithms on the PPI network
  - Embedded within two size-4 cliques
  - Also part of many other size-3 cliques
- After weighting by SSS, the internal weights of the complex remain high, while extraneous weights are lowered \(\rightarrow\) Can be found in all cross-validation rounds
Human ubiquitin ligase

- Two human ubiquitin ligase complexes, which share 1 protein in common (UBE2N)
-Cannot be found by standard clustering algorithms on the PPI network
  - Embedded within many larger cliques
  - Many extraneous edges
- After weighting by SSS, the internal weights of the complex remain high, while extraneous weights are lowered
  - UBE2V2-UBE2N can be found in all cross-validation rounds
  - UBE2V1-UBE2N can be found in 78% of cross-validation rounds
Conclusion

• **Most complexes are small, so small-complex prediction is an impt part of complex prediction**

• **Many challenges in small-complex prediction**
  – Searching for dense clusters is ineffectual
  – Sensitive to noise
  – Scoring candidate complexes is sensitive to edge weights

• **SSS + Extract**
  – Integrate 3 data sources w/ their topological features
  – Size-specific edge weighting by supervised learning
  – When scoring candidate complexes, incorporates outgoing edges from clusters as well

  ⇒ **Much improved performance in yeast and human**
Must read


Good to read


Acknowledgements

• A lot of the slides for this lecture were adapted from ppt files given to me by Sriganesh Srihari and Yong Chern Han

• A lot of the results presented here are from the work of Liu Guimei, Yong Chern Han, and Sriganesh Srihari