Discovering Binding Motif Pairs from Interacting Protein Groups

Limsoon Wong
Institute for Infocomm Research
Singapore

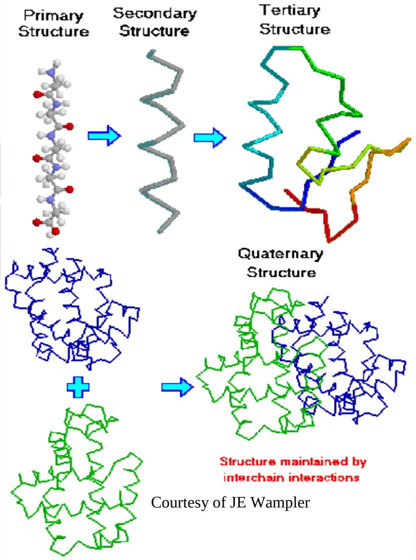
Plan

- Motivation from biology & problem statement
- Recasting as a graph theory problem
- Recasting as a data mining problem
- Mining interacting protein groups
- Generating motif pairs
- Results and validation

Motivation from Biology

Proteins

- 4 types of reps for proteins: primary, secondary, tertiary, & quaternay
- Protein interactions play impt role in inter cellular communication, in signal transduction, & in the regulation of gene expression

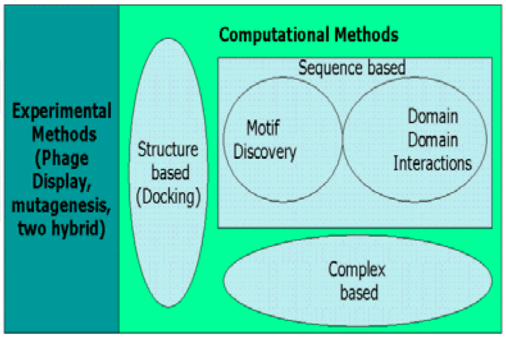


Courtesy of JE Wampler

Binding Sites

- Discovery of binding sites is a key part of understanding mechanisms of protein interactions
- Structure-based approaches
  - E.g., docking
  - Relatively accurate
  - Struct must be known

⇒ Sequence-based approaches



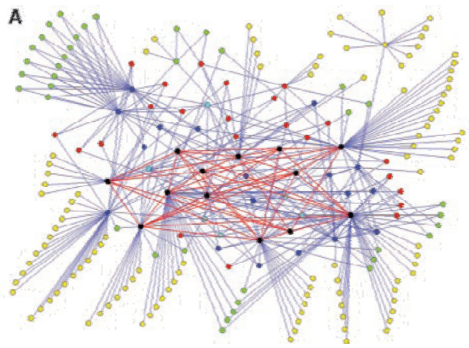
Typical Sequence-Based Approach

- Typical sequence-based approaches have two steps:
  - Use pattern discovery algorithms to discover domains and/or motifs of a group of proteins
  - Use domain-domain interaction discovery methods (e.g., domain fusion) to discovery interacting domains
- Shortcomings:
  - Protein interaction information is not used by motif discovery algorithms
  - Exact positions of binding sites often not recognized

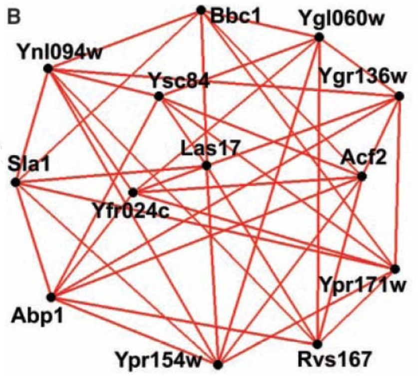
How about ...

- How about making use of known protein-protein bindings to guide the discovery of binding motifs?

Protein Interaction Graphs

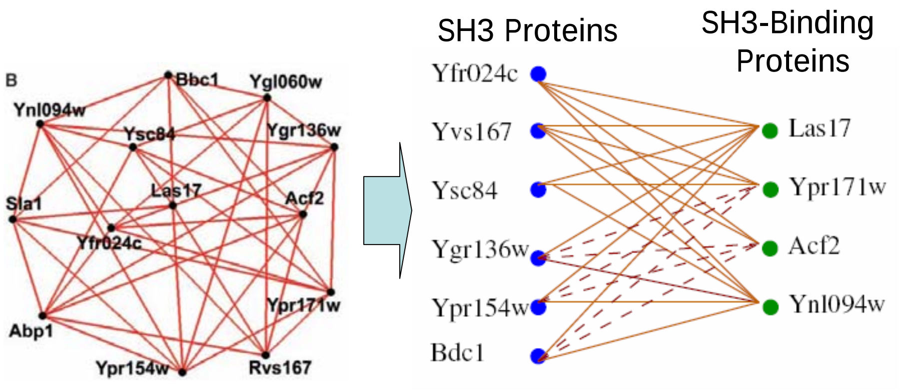


Yeast SH3 domain-domain Interaction network:
394 edges, 206 nodes
Tong et al. *Science*, v295. 2002



8 proteins containing SH3
5 binding at least 6 of them

Bipartite Subgraphs



The larger this group, the more likely their active sites will show up clearly in a multiple alignment?

Problem Statement

Given a PPI expt E, the problem is

(1) To find all pairs X, Y of **interacting protein groups**, so that

- (1.1) every protein in X interacts with every protein in Y, &
- (1.2) X and Y are as large as possible

&

(2) To identify “good” binding motif pairs from these pairs of interacting protein groups

Recasting As a Graph Theory Problem

PPI Expt As a Graph

- PPI expt E as undirected graph $G^E = \langle V^E, D^E \rangle$,
  - where V^E are the proteins and D^E the edges,
  - so that two proteins are connected in G^E iff there is a binding betw them in PPI expt E
- Let $L^E(p)$ denote the **neighborhood** of protein p in G^E
- Let $L^E(P) = \bigcap_{p \in P} L^E(p)$ denote the common neighborhood of all proteins in P in G^E

Maximality

- Proposition 2.1

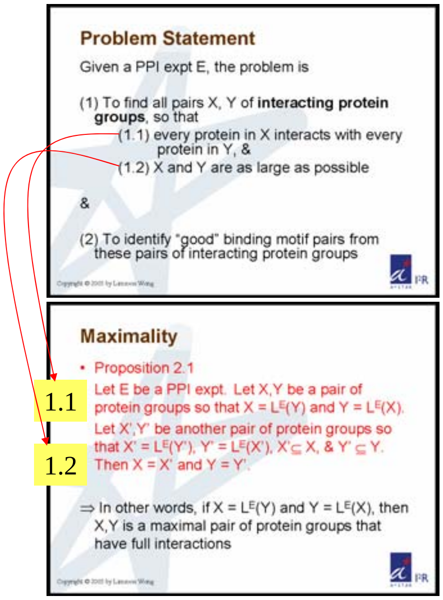
  Let E be a PPI expt. Let X,Y be a pair of protein groups so that $X = L^E(Y)$ and $Y = L^E(X)$.

  Let X',Y' be another pair of protein groups so that $X' = L^E(Y')$, $Y' = L^E(X')$, $X' \subseteq X$, & $Y' \subseteq Y$. Then $X = X'$ and $Y = Y'$.

$\Rightarrow$ In other words, if $X = L^E(Y)$ and $Y = L^E(X)$, then X,Y is a maximal pair of protein groups that have full interactions

---

Recasting to Graph Theory



Problem Statement

Given a PPI expt E, the problem is

(1) To find all pairs X, Y of **interacting protein groups**, so that
  (1.1) every protein in X interacts with every protein in Y, &
  (1.2) X and Y are as large as possible

&

(2) To identify "good" binding motif pairs from these pairs of interacting protein groups

Maximality

- Proposition 2.1

  Let E be a PPI expt. Let X,Y be a pair of protein groups so that $X = L^E(Y)$ and $Y = L^E(X)$.

  Let X',Y' be another pair of protein groups so that $X' = L^E(Y')$, $Y' = L^E(X')$, $X' \subseteq X$, & $Y' \subseteq Y$. Then $X = X'$ and $Y = Y'$.

$\Rightarrow$ In other words, if $X = L^E(Y)$ and $Y = L^E(X)$, then X,Y is a maximal pair of protein groups that have full interactions

Copyright © 2005 by Limsoon Wong

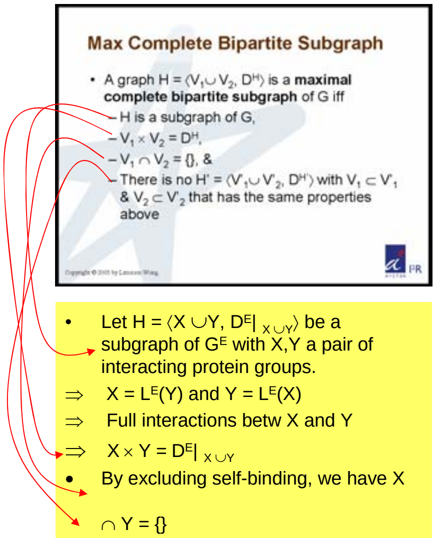
- X, Y is a pair of interacting protein groups in PPI expt E iff $X = L^E(Y)$ and $Y = L^E(X)$

Max Complete Bipartite Subgraph

- A graph $H = \langle V_1 \cup V_2, D^H \rangle$ is a **maximal complete bipartite subgraph** of G iff
  - H is a subgraph of G,
  - $V_1 \times V_2 = D^H$,
  - $V_1 \cap V_2 = \{\}$, &
  - There is no $H' = \langle V'_1 \cup V'_2, D^{H'} \rangle$ with $V_1 \subset V'_1$ & $V_2 \subset V'_2$ that has the same properties above

---

The Connection to Graph Theory


Max Complete Bipartite Subgraph

- A graph $H = \langle V_1 \cup V_2, D^H \rangle$ is a **maximal complete bipartite subgraph** of G iff
  - H is a subgraph of G,
  - $V_1 \times V_2 = D^H$,
  - $V_1 \cap V_2 = \{\}$, &
  - There is no $H' = \langle V'_1 \cup V'_2, D^{H'} \rangle$ with $V_1 \subset V'_1$ & $V_2 \subset V'_2$ that has the same properties above

Copyright © 2005 by Limsoon Wong.

- Let $H = \langle X \cup Y, D^E|_{X \cup Y} \rangle$ be a subgraph of G^E with X,Y a pair of interacting protein groups.

⇒ $X = L^E(Y)$ and $Y = L^E(X)$

⇒ Full interactions betw X and Y

⇒ $X \times Y = D^E|_{X \cup Y}$

- By excluding self-binding, we have $X \cap Y = \{\}$
- By Proposition 2.1, we have H is max

- X, Y is a pair of interacting protein groups in PPI expt E iff $H = \langle X \cup Y, X \times Y \rangle$ is max complete bipartite subgraph of G^E

Therefore … But ...

- Therefore, to find pairs of interacting protein groups, we can use algorithms from graph theory for enumerating maximal complete bipartite subgraphs
- According to Eppstein 1994, this has complexity $O(a^3 2^{2a} n)$, where a is the aboricity of the graph and n the number of vertices
- This is inefficient because a is often around 10-20 in practice

---

Recasting As a Data Mining Problem

From PPI Expts To Transactions

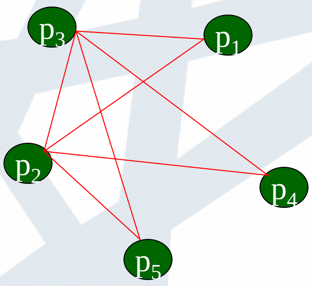
- In PPI expt E, we obtain for each protein p, a list $L^E(p)$ of proteins that bind p
  - – assume $p \notin L^E(p)$, as such expts are not intended to detect self-binding
  - – assume $q \in L^E(p)$ implies $p \in L^E(q)$, as binding is symmetric
- $L^E(p)$ can be thought of as a transaction & $t^E(p)$ as the "id" of this transaction

⇒ E can be thought of as generating a **db of transactions** $D^E = \{t^E(p_1), \ldots, t^E(p_k)\}$, where $p_1, \ldots, p_k$ are all the proteins involved in E

⇒ a set of proteins X can be thought of as a **pattern** in D^E if there is $t^E(p) \in D^E$ st $X \subseteq L^E(p)$

Example

- Consider expt E with 5 proteins $p_1, \ldots, p_5$, st p_2 and p_3 bind every protein except themselves
- Then D^E looks like this (as a matrix):



	p_1	p_2	p_3	p_4	p_5
$t(p_1)$	0	1	1	0	0
$t(p_2)$	1	0	1	1	1
$t(p_3)$	1	1	0	1	1
$t(p_4)$	0	1	1	0	0
$t(p_5)$	0	1	1	0	0

Notations

- Let $s^E(d)$ denote the protein p st $t^E(p) = d$

$\Rightarrow$ $s^E(t^E(p)) = p$

- Let $t^E(X)$ denote the set $\{t^E(p) \mid p \in X\}$ of transaction id's, where X is a pattern in D^E
- Let $s^E(T)$ denote the pattern $\{s^E(d) \mid d \in T\}$
- Let $[|p|]^E$ denote the set $\{t^E(q) \mid p \in L^E(q)\}$ of transactions in D^E in which p occurs

$\Rightarrow$ $t^E(p) \in [|q|]^E$ implies $t^E(q) \in [|p|]^E$

- Let $[|X|]^E$ denote the set $\bigcap_{p \in X} [|p|]^E$ of transactions in which the pattern X occurs

$\Rightarrow$ $t^E(Y) \subseteq [|X|]^E$ implies $t^E(X) \subseteq [|Y|]^E$

Closed Patterns

- Let $[X]^E = \{Y \mid [|Y|]^E = [|X|]^E\}$ denote the **equivalence class** of the pattern X in D^E
- A pattern X is said to be a **closed pattern** of D^E iff $X = \text{closed}^E(X)$, where $\{\text{closed}^E(X)\} = \max [X]^E$

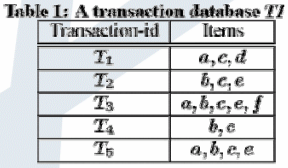
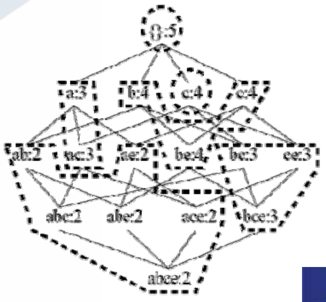


Table 1: A transaction database *TI*

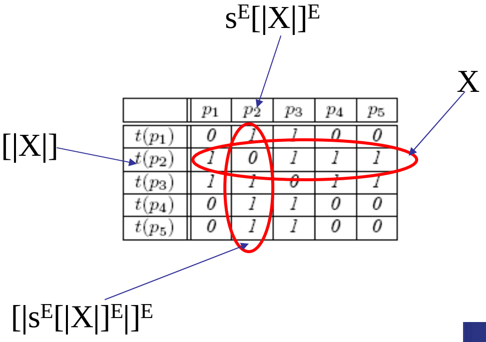
Transaction-id	Items
T_1	a, c, d
T_2	b, c, e
T_3	a, b, c, e, f
T_4	b, c
T_5	a, b, c, e



Support threshold = 2

Key Proposition

- Proposition 3.2
  Let X be a closed pattern in D^E.
  Then $X = s^E\,[|s^E\,[|X|]^E\,|]^E$



$s^E[|X|]^E$

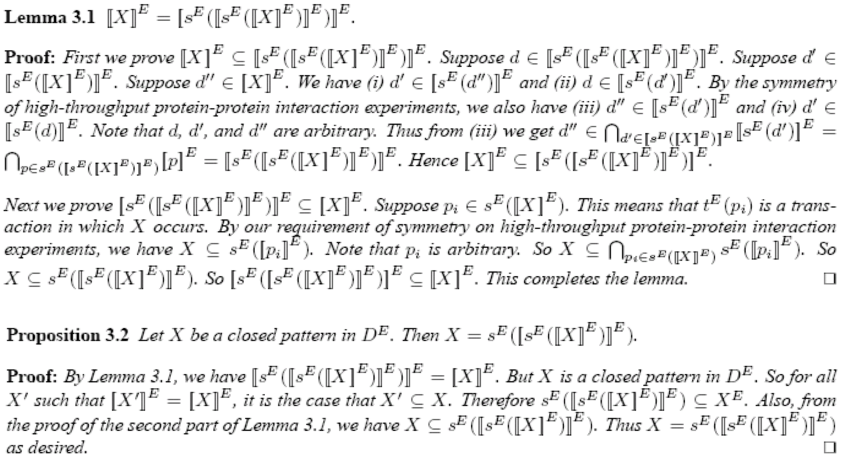
X

$[|X|]$

	p_1	p_2	p_3	p_4	p_5
$t(p_1)$	0	1	1	0	0
$t(p_2)$	1	0	1	1	1
$t(p_3)$	1	1	0	1	1
$t(p_4)$	0	1	1	0	0
$t(p_5)$	0	1	1	0	0

$[|s^E[|X|]^E|]^E$

Proof



Lemma 3.1 $\llbracket X\rrbracket^E = [s^E(\llbracket s^E(\llbracket X\rrbracket^E)\rrbracket^E)\rrbracket^E$.

Proof: *First we prove* $\llbracket X\rrbracket^E \subseteq \llbracket s^E(\llbracket s^E(\llbracket X\rrbracket^E)\rrbracket^E)\rrbracket^E$. *Suppose* $d \in \llbracket s^E(\llbracket s^E(\llbracket X\rrbracket^E)\rrbracket^E)\rrbracket^E$. *Suppose* $d' \in \llbracket s^E(\llbracket X\rrbracket^E)\rrbracket^E$. *Suppose* $d'' \in [X\rrbracket^E$. *We have (i)* $d' \in [s^E(d'')\rrbracket^E$ *and (ii)* $d \in \llbracket s^E(d')\rrbracket^E$. *By the symmetry of high-throughput protein-protein interaction experiments, we also have (iii)* $d'' \in \llbracket s^E(d')\rrbracket^E$ *and (iv)* $d' \in \llbracket s^E(d)\rrbracket^E$. *Note that* d, d', *and* d'' *are arbitrary. Thus from (iii) we get* $d'' \in \bigcap_{d' \in [s^E([X]^E)]^E} \llbracket s^E(d')\rrbracket^E = \bigcap_{p \in s^E([s^E([X]^E)]^E)} [p\rrbracket^E = \llbracket s^E(\llbracket s^E(\llbracket X\rrbracket^E)\rrbracket^E)\rrbracket^E$. *Hence* $[X\rrbracket^E \subseteq [s^E([s^E(\llbracket X\rrbracket^E)\rrbracket^E)]^E$.

Next we prove $[s^E(\llbracket s^E(\llbracket X\rrbracket^E)\rrbracket^E)\rrbracket^E \subseteq [X\rrbracket^E$. *Suppose* $p_i \in s^E(\llbracket X\rrbracket^E)$. *This means that* $t^E(p_i)$ *is a transaction in which* X *occurs. By our requirement of symmetry on high-throughput protein-protein interaction experiments, we have* $X \subseteq s^E([p_i\rrbracket^E)$. *Note that* p_i *is arbitrary. So* $X \subseteq \bigcap_{p_i \in s^E(\llbracket X\rrbracket^E)} s^E(\llbracket p_i\rrbracket^E)$. *So* $X \subseteq s^E(\llbracket s^E(\llbracket X\rrbracket^E)\rrbracket^E)$. *So* $[s^E([s^E(\llbracket X\rrbracket^E)\rrbracket^E)]^E \subseteq \llbracket X\rrbracket^E$. *This completes the lemma.* $\square$

Proposition 3.2 *Let* X *be a closed pattern in* D^E. *Then* $X = s^E([s^E(\llbracket X\rrbracket^E)\rrbracket^E)$.

Proof: *By Lemma 3.1, we have* $\llbracket s^E(\llbracket s^E(\llbracket X\rrbracket^E)\rrbracket^E)\rrbracket^E = [X\rrbracket^E$. *But* X *is a closed pattern in* D^E. *So for all* X' *such that* $[X'\rrbracket^E = [X\rrbracket^E$, *it is the case that* $X' \subseteq X$. *Therefore* $s^E(\llbracket s^E(\llbracket X\rrbracket^E)\rrbracket^E) \subseteq X^E$. *Also, from the proof of the second part of Lemma 3.1, we have* $X \subseteq s^E(\llbracket s^E(\llbracket X\rrbracket^E)\rrbracket^E)$. *Thus* $X = s^E(\llbracket s^E(\llbracket X\rrbracket^E)\rrbracket^E)$ *as desired.* $\square$

Consequently...

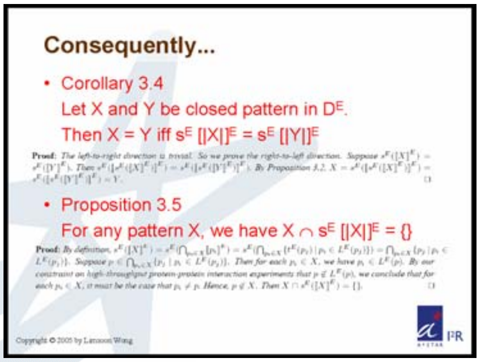
- Corollary 3.4
  Let X and Y be closed pattern in D^E.
  Then X = Y iff $s^E [|X|]^E = s^E [|Y|]^E$

Proof: *The left-to-right direction is trivial. So we prove the right-to-left direction. Suppose* $s^E([\![X]\!]^E) = s^E([\![Y]\!]^E)$. *Then* $s^E([\![s^E([\![X]\!]^E)]\!]^E) = s^E([\![s^E([\![Y]\!]^E)]\!]^E)$. *By Proposition 3.2,* $X = s^E([\![s^E([\![X]\!]^E)]\!]^E) = s^E([\![s^E([\![Y]\!]^E)]\!]^E) = Y$. □

- Proposition 3.5
  For any pattern X, we have $X \cap s^E [|X|]^E = \{\}$

Proof: *By definition,* $s^E([\![X]\!]^E) = s^E(\bigcap_{p_i \in X} [\![p_i]\!]^E) = s^E(\bigcap_{p_i \in X} \{t^E(p_j) \mid p_i \in L^E(p_j)\}) = \bigcap_{p_i \in X} \{p_j \mid p_i \in L^E(p_j)\}$. *Suppose* $p \in \bigcap_{p_i \in X} \{p_j \mid p_i \in L^E(p_j)\}$. *Then for each* $p_i \in X$, *we have* $p_i \in L^E(p)$. *By our constraint on high-throughput protein-protein interaction experiments that* $p \notin L^E(p)$, *we conclude that for each* $p_i \in X$, *it must be the case that* $p_i \neq p$. *Hence,* $p \notin X$. *Then* $X \cap s^E([\![X]\!]^E) = \{\}$. □

Implication



Consequently...

- Corollary 3.4
  Let X and Y be closed pattern in D^E.
  Then X = Y iff $s^E [|X|]^E = s^E [|Y|]^E$
- Proposition 3.5
  For any pattern X, we have $X \cap s^E [|X|]^E = \{\}$

Copyright © 2005 by Limsoon Wong.

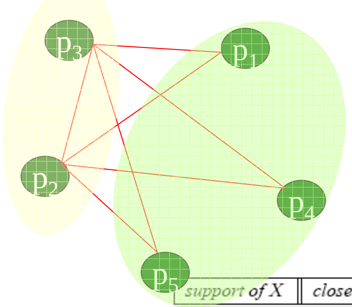
- Corollary 3.6
  Let E be a PPI expt.
  Let C be the set of closed patterns of D^E.
  Then |C| is even

Proof: *By Corollary 3.4,* $s^E \circ [\![\cdot]\!]^E$ *is a bijective function from* C *to* C. *Thus, if* $|C|$ *is odd, there must be some* $X \in C$ *such that* $X = s^E([\![X]\!]^E)$. *But this contradicts Proposition 3.5. So* $|C|$ *must be even.* □

More Important Implication

- Theorem 3.7

  The bijective function $s^E o[|\cdot|]^E$ partitions the set of closed patterns of D^E into bipartite graphs



	p_1	p_2	p_3	p_4	p_5
$t(p_1)$	0	1	1	0	0
$t(p_2)$	1	0	1	1	1
$t(p_3)$	1	1	0	1	1
$t(p_4)$	0	1	1	0	0
$t(p_5)$	0	1	1	0	0

support of X	*close pattern X*	$Y = s^E([\![X]\!]^E)$	*support of Y*
3	$\{p_2, p_3\}$	$\{p_1, p_4, p_5\}$	2
4	$\{p_2\}$	$\{p_1, p_3, p_4, p_5\}$	1
4	$\{p_3\}$	$\{p_1, p_2, p_4, p_5\}$	1

Even More Interesting...

- For each pair X and Y,
  - X is the largest group of proteins that bind all the proteins in Y; and
  - Y is the largest group of proteins that bind all the proteins in X

⇒ X,Y is a pair of interacting protein groups

	p_1	p_2	p_3	p_4	p_5
$t(p_1)$	0	1	1	0	0
$t(p_2)$	1	0	1	1	1
$t(p_3)$	1	1	0	1	1
$t(p_4)$	0	1	1	0	0
$t(p_5)$	0	1	1	0	0

support of X	*close pattern X*	$Y = s^E([\![X]\!]^E)$	*support of Y*
3	$\{p_2, p_3\}$	$\{p_1, p_4, p_5\}$	2
4	$\{p_2\}$	$\{p_1, p_3, p_4, p_5\}$	1
4	$\{p_3\}$	$\{p_1, p_2, p_4, p_5\}$	1

A Couple More Propositions

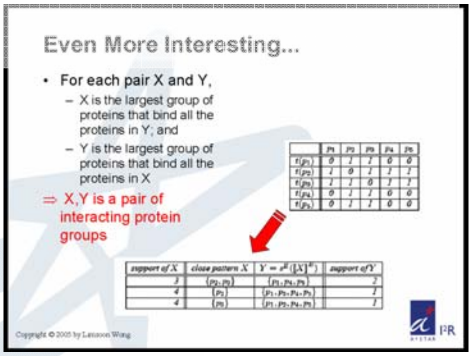
- Proposition 3.3
  For any pattern X,
  $s^E[|X|]^E$ is closed pattern in D^E

Proof: *Let* $Y = closed(s^E([\![X]\!]^E))$. *Then* $[\![Y]\!]^E = [\![s^E([\![X]\!]^E)]\!]^E$. *Then* $s^E([\![Y]\!]^E) = s^E([\![s^E([\![X]\!]^E)]\!]^E)$. *Then* $s^E([\![s^E([\![Y]\!]^E)]\!]^E) = s^E([\![s^E([\![s^E([\![X]\!]^E)]\!]^E)]\!]^E)$. *Since* Y *is a closed pattern in* D^E, *by Proposition 3.2 and Lemma 3.1,* $Y = s^E([\![s^E([\![Y]\!]^E)]\!]^E) = s^E([\![s^E([\![s^E([\![X]\!]^E)]\!]^E)]\!]^E) = s^E([\![X]\!]^E)$. *Hence* $s^E([\![X]\!]^E)$ *is also a closed pattern in* D^E. □

- Corollary 3.7
  X is a closed pattern in D^E iff $X = s^E[|s^E[|X|]^E|]^E$

Proof: *This follows directly from Proposition 3.2 and Proposition 3.3.* □

At Last!



Even More Interesting...

- For each pair X and Y,
  - X is the largest group of proteins that bind all the proteins in Y; and
  - Y is the largest group of proteins that bind all the proteins in X

⇒ X,Y is a pair of interacting protein groups

	p_1	p_2	p_3	p_4	p_5
$f(p_1)$	0	1	1	0	0
$f(p_2)$	1	0	1	1	1
$f(p_3)$	1	1	0	1	1
$f(p_4)$	0	1	1	0	0
$f(p_5)$	0	1	1	0	0

support of X	close pattern X	$Y = s^E([\![X]\!]^E)$	support of Y
3	$\{p_2, p_3\}$	$\{p_1, p_4, p_5\}$	2
4	$\{p_2\}$	$\{p_1, p_3, p_4, p_5\}$	1
4	$\{p_3\}$	$\{p_1, p_2, p_4, p_5\}$	1

Copyright © 2005 by Limsoon Wong

- These are ALL the interacting protein groups

⇒ To mine these protein groups, it suffices to mine closed patterns in D^E

An Extension

- Not all interacting protein groups X, Y are equally interesting
  - – X and Y are both singleton, vs
  - – X is a large group, Y is small group, vs
  - – X is a large group, Y is a large group

⇒ Set “interestingness” threshold on X, Y st a pair of interacting protein groups X, Y is interesting only if $|X| \geq m$ and $|Y| \geq n$

An Optimization

- Let X, Y be a pair of interacting protein groups
  - – By Theorem 3.7, $X = s^E [|Y|]^E$ and $Y = s^E [|X|]^E$
  - – By Definition of $[|\bullet|]^E$, $|X|$ = times Y occurs in D^E
  - – By Definition of $[|\bullet|]^E$, $|Y|$ = times X occurs in D^E

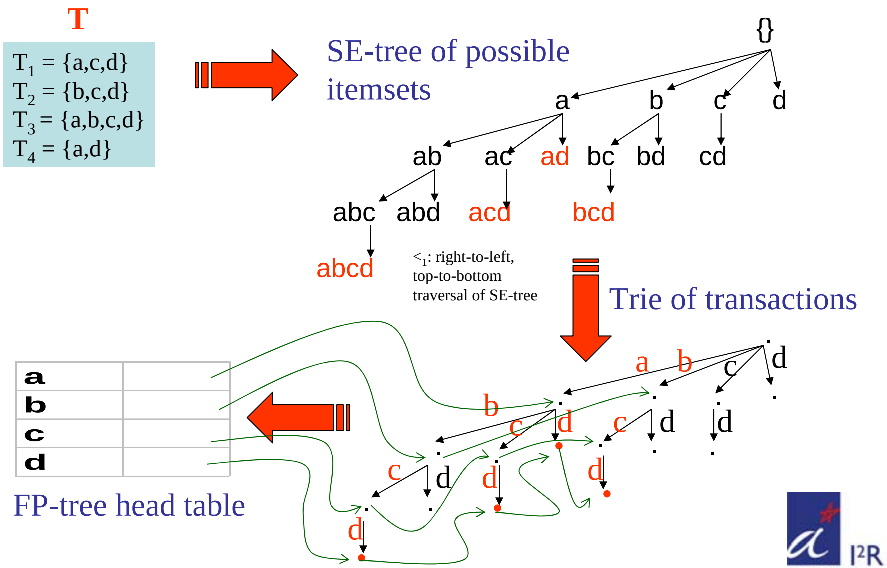
⇒ To mine interesting pairs X, Y of interacting protein groups in an expt E such that $|X| \geq m$ and $|Y| \geq n$, it suffices to mine closed patterns X that appears $\geq n$ times in D^E and $|X| \geq m$

Mining Closed Patterns Efficiently

Closed Pattern Mining Algorithms

- CLOSET, Pei et al. 2000
- CARPENTER, Pan et al. 2003
- FPclose*, Grahne & Zhu 2003
- GC-growth, Li et al. 2005
- ...

From SE-Tree To Trie To FP-Tree



T

$T_1 = \{a,c,d\}$
$T_2 = \{b,c,d\}$
$T_3 = \{a,b,c,d\}$
$T_4 = \{a,d\}$

SE-tree of possible itemsets

$<_1$: right-to-left, top-to-bottom traversal of SE-tree

Trie of transactions

FP-tree head table

GC-growth: Fast Simultaneous Mining of Generators & Closed Patterns

- $P[X] = |\{d_T \mid T \in \mathcal{D}, X \text{ is a prefix of } T\}|$;
- $P^{pos}[X] = |\{d_T \mid T \in \mathcal{D}^{pos}, X \text{ is a prefix of } T\}|$;
- $P^{neg}[X] = |\{d_T \mid T \in \mathcal{D}^{neg}, X \text{ is a prefix of } T\}|$;
- $S[X] = sup(X, \mathcal{D})$;
- $S^{pos}[X] = sup(X, \mathcal{D}^{pos})$;
- $S^{neg}[X] = sup(X, \mathcal{D}^{neg})$;
- $G[X] = $ true iff X is a generator in $\mathcal{D}$; and
- $H[\alpha] = \{X \mid P[X] \text{ is defined}, \{\alpha\} \text{ is suffix of } X\}$.

```
Input: Dataset $\mathcal{D} = \mathcal{D}^{pos} \cup \mathcal{D}^{neg}$, and support threshold $ms$.
Output: The generators and closed patterns of $\mathcal{F}(ms, \mathcal{D})$, as well as and their support levels.
Method:
Fill in $P[\cdot]$, $P^{pos}[\cdot]$, $P^{neg}[\cdot]$, and $H[\cdot]$ by making one pass through $\mathcal{D}$; set $G[\{\}] :=$ true; initialize $R[\cdot]$ to empty.
Let $X_1$, ..., $X_n$ be the nodes in the set-enumeration tree on the possible itemsets in $\mathcal{D}$, where $X_1 <_1 X_2 <_1 \cdots <_1 X_n$.
for $i = 1, .., n$ do
    $S[X_i] := \sum_{X \in H[last(X_i)], X_i \subseteq X} P[X]$;
    if $S[X_i] \geq ms$ and $\forall \alpha \in X_i.S[X_i] < S[X_i - \{\alpha\}]$ and $\forall \alpha \in X_i.G[X_i - \{\alpha\}] =$ true then
        $G[X_i] :=$ true;
        $C := \bigcap \max\{X'' \mid X' \in H[last(X_i)], X_i \subseteq X', X'$ is prefix of $X'', P[X''] \leq P[X']\}$;
        $R[C] := R[C] \cup \{X_i\}$;
        $S^{pos}[C] := \sum_{X \in H[last(X_i)], X_i \subseteq X} P^{pos}[X]$;
        $S^{neg}[C] := \sum_{X \in H[last(X_i)], X_i \subseteq X} P^{neg}[X]$;
    else
        $G[X_i] :=$ false;
    end if
end for
for each $C$ such that $R[C] \neq \{\}$ do
    output $R[C]$, $C$, $S^{pos}[C]$, and $S^{neg}[C]$;
end for
```

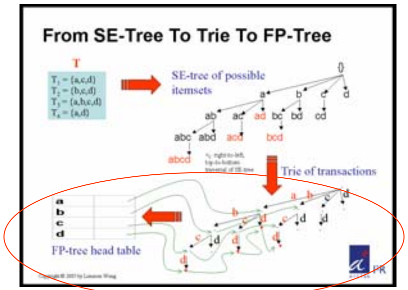
Step 1: FP-tree construction

Input: Dataset $\mathcal{D} = \mathcal{D}^{pos} \cup \mathcal{D}^{neg}$, and support threshold ms.

Output: The generators and closed patterns of $\mathcal{F}(ms, \mathcal{D})$, as well as and their support levels.

Method:

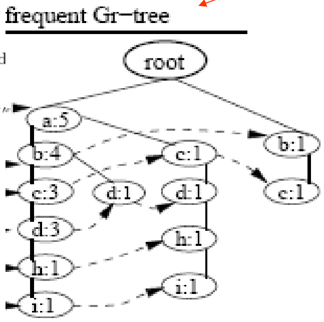
```
Fill in $P[\cdot]$, $P^{pos}[\cdot]$, $P^{neg}[\cdot]$, and $H[\cdot]$ by making one pass through $\mathcal{D}$; set $G[\{\}] :=$ true; initialize $R[\cdot]$ to empty.
Let $X_1$, ..., $X_n$ be the nodes in the set-enumeration tree on the possible itemsets in $\mathcal{D}$, where $X_1 <_1 X_2 <_1 \cdots <_1 X_n$.
for $i = 1, .., n$ do
    $S[X_i] := \sum_{X \in H[last(X_i)], X_i \subseteq X} P[X]$;
    if $S[X_i] \geq ms$ and $\forall \alpha \in X_i . S[X_i] < S[X_i - \{\alpha\}]$ and $\forall \alpha \in X_i . G[X_i - \{\alpha\}] =$ true then
        $G[X_i] :=$ true;
        $C := \bigcap \max\{X'' \mid X' \in H[last(X_i)], X_i \subseteq X', X''$ is prefix of $X''$, $P[X''] \leq P[X']\}$;
        $R[C] := R[C] \cup \{X_i\}$;
        $S^{pos}[C] := \sum_{X \in H[last(X_i)], X_i \subseteq X} P^{pos}[X]$;
        $S^{neg}[C] := \sum_{X \in H[last(X_i)], X_i \subseteq X} P^{neg}[X]$;
    else
        $G[X_i] :=$ false;
    end if
end for
for each $C$ such that $R[C] \neq \{\}$ do
    output $R[C]$, $C$, $S^{pos}[C]$, and $S^{neg}[C]$;
end for
```



From SE-Tree To Trie To FP-Tree

T

T_1 = {a,c,d}
T_2 = {b,c,d}
T_3 = {a,b,c,d}
T_4 = {a,d}



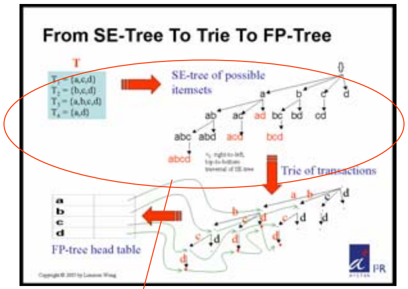
Step 2: Right-to-left, top-to-bottom traversal

Input: Dataset $\mathcal{D} = \mathcal{D}^{pos} \cup \mathcal{D}^{neg}$, and support threshold ms.

Output: The generators and closed patterns of $\mathcal{F}(ms, \mathcal{D})$, as well as and their support levels.

Method:

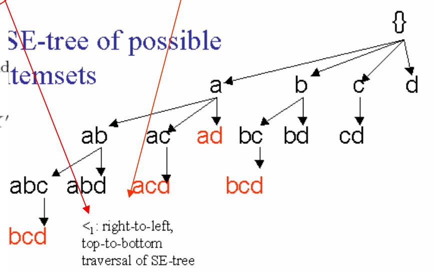
```
Fill in $P[\cdot]$, $P^{pos}[\cdot]$, $P^{neg}[\cdot]$, and $H[\cdot]$ by making one pass through $\mathcal{D}$; set $G[\{\}] :=$ true; initialize $R[\cdot]$ to empty.
Let $X_1$, ..., $X_n$ be the nodes in the set-enumeration tree on the possible itemsets in $\mathcal{D}$, where $X_1 <_1 X_2 <_1 \cdots <_1 X_n$.
for $i = 1, .., n$ do
    $S[X_i] := \sum_{X \in H[last(X_i)], X_i \subseteq X} P[X]$;
    if $S[X_i] \geq ms$ and $\forall \alpha \in X_i . S[X_i] < S[X_i - \{\alpha\}]$ and $\forall \alpha \in X_i . G[X_i - \{\alpha\}] =$ true then
        $G[X_i] :=$ true;
        $C := \bigcap \max\{X'' \mid X' \in H[last(X_i)], X_i \subseteq X', X'$ is prefix of $X''$, $P[X''] \leq P[X']\}$;
        $R[C] := R[C] \cup \{X_i\}$;
        $S^{pos}[C] := \sum_{X \in H[last(X_i)], X_i \subseteq X} P^{pos}[X]$;
        $S^{neg}[C] := \sum_{X \in H[last(X_i)], X_i \subseteq X} P^{neg}[X]$;
    else
        $G[X_i] :=$ false;
    end if
end for
for each $C$ such that $R[C] \neq \{\}$ do
    output $R[C]$, $C$, $S^{pos}[C]$, and $S^{neg}[C]$;
end for
```

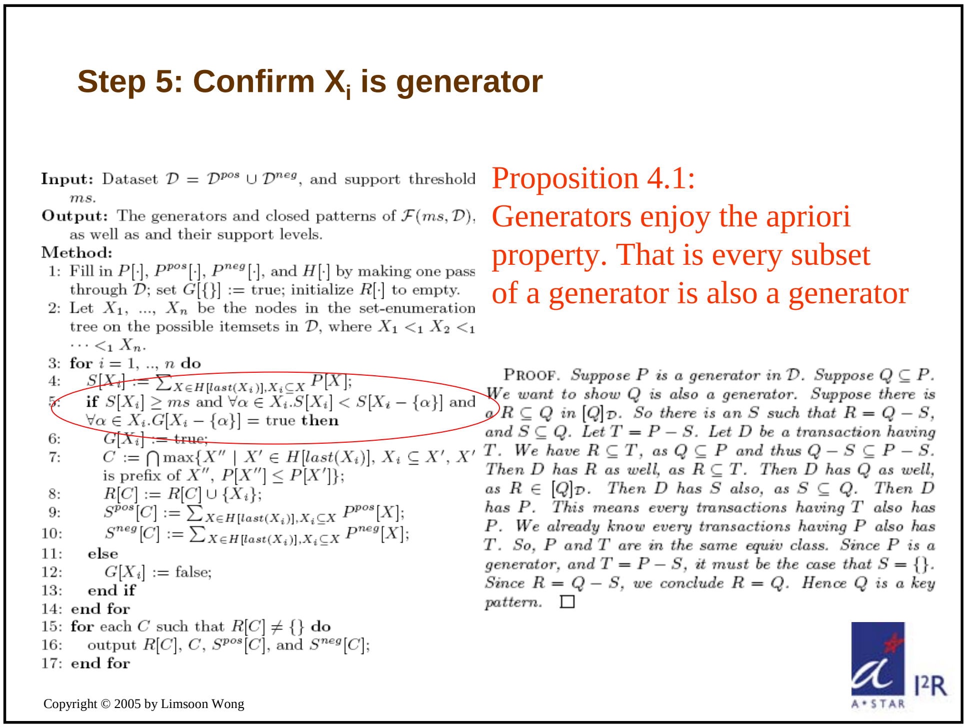


From SE-Tree To Trie To FP-Tree

T

T_1 = {a,c,d}
T_2 = {b,c,d}
T_3 = {a,b,c,d}
T_4 = {a,d}



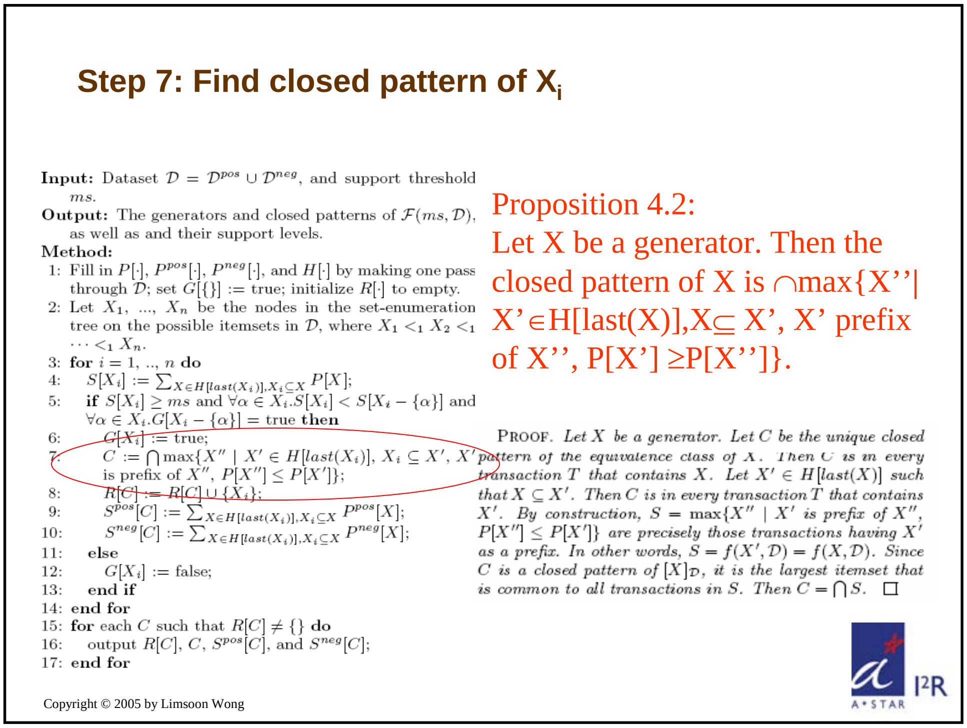


Step 5: Confirm X_i is generator

```
Input: Dataset $\mathcal{D} = \mathcal{D}^{pos} \cup \mathcal{D}^{neg}$, and support threshold $ms$.
Output: The generators and closed patterns of $\mathcal{F}(ms, \mathcal{D})$, as well as and their support levels.
Method:
1: Fill in $P[\cdot]$, $P^{pos}[\cdot]$, $P^{neg}[\cdot]$, and $H[\cdot]$ by making one pass through $\mathcal{D}$; set $G[\{\}] :=$ true; initialize $R[\cdot]$ to empty.
2: Let $X_1$, ..., $X_n$ be the nodes in the set-enumeration tree on the possible itemsets in $\mathcal{D}$, where $X_1 <_1 X_2 <_1 \cdots <_1 X_n$.
3: for $i = 1, .., n$ do
4:   $S[X_i] := \sum_{X \in H[last(X_i)], X_i \subseteq X} P[X]$;
5:   if $S[X_i] \geq ms$ and $\forall \alpha \in X_i . S[X_i] < S[X_i - \{\alpha\}]$ and $\forall \alpha \in X_i . G[X_i - \{\alpha\}] =$ true then
6:     $G[X_i] :=$ true;
7:     $C := \bigcap \max\{X'' \mid X' \in H[last(X_i)], X_i \subseteq X', X'$ is prefix of $X''$, $P[X''] \leq P[X']\}$;
8:     $R[C] := R[C] \cup \{X_i\}$;
9:     $S^{pos}[C] := \sum_{X \in H[last(X_i)], X_i \subseteq X} P^{pos}[X]$;
10:    $S^{neg}[C] := \sum_{X \in H[last(X_i)], X_i \subseteq X} P^{neg}[X]$;
11:  else
12:    $G[X_i] :=$ false;
13:  end if
14: end for
15: for each $C$ such that $R[C] \neq \{\}$ do
16:   output $R[C]$, $C$, $S^{pos}[C]$, and $S^{neg}[C]$;
17: end for
```

Proposition 4.1:
Generators enjoy the apriori property. That is every subset of a generator is also a generator

PROOF. *Suppose P is a generator in $\mathcal{D}$. Suppose $Q \subseteq P$. We want to show Q is also a generator. Suppose there is a $R \subseteq Q$ in $[Q]_{\mathcal{D}}$. So there is an S such that $R = Q - S$, and $S \subseteq Q$. Let $T = P - S$. Let D be a transaction having T. We have $R \subseteq T$, as $Q \subseteq P$ and thus $Q - S \subseteq P - S$. Then D has R as well, as $R \subseteq T$. Then D has Q as well, as $R \in [Q]_{\mathcal{D}}$. Then D has S also, as $S \subseteq Q$. Then D has P. This means every transactions having T also has P. We already know every transactions having P also has T. So, P and T are in the same equiv class. Since P is a generator, and $T = P - S$, it must be the case that $S = \{\}$. Since $R = Q - S$, we conclude $R = Q$. Hence Q is a key pattern.* □



Step 7: Find closed pattern of X_i

```
Input: Dataset $\mathcal{D} = \mathcal{D}^{pos} \cup \mathcal{D}^{neg}$, and support threshold $ms$.
Output: The generators and closed patterns of $\mathcal{F}(ms, \mathcal{D})$, as well as and their support levels.
Method:
1: Fill in $P[\cdot]$, $P^{pos}[\cdot]$, $P^{neg}[\cdot]$, and $H[\cdot]$ by making one pass through $\mathcal{D}$; set $G[\{\}] :=$ true; initialize $R[\cdot]$ to empty.
2: Let $X_1$, ..., $X_n$ be the nodes in the set-enumeration tree on the possible itemsets in $\mathcal{D}$, where $X_1 <_1 X_2 <_1 \cdots <_1 X_n$.
3: for $i = 1, .., n$ do
4:   $S[X_i] := \sum_{X \in H[last(X_i)], X_i \subseteq X} P[X]$;
5:   if $S[X_i] \geq ms$ and $\forall \alpha \in X_i . S[X_i] < S[X_i - \{\alpha\}]$ and $\forall \alpha \in X_i . G[X_i - \{\alpha\}] =$ true then
6:     $G[X_i] :=$ true;
7:     $C := \bigcap \max\{X'' \mid X' \in H[last(X_i)], X_i \subseteq X', X'$ is prefix of $X''$, $P[X''] \leq P[X']\}$;
8:     $R[C] := R[C] \cup \{X_i\}$;
9:     $S^{pos}[C] := \sum_{X \in H[last(X_i)], X_i \subseteq X} P^{pos}[X]$;
10:    $S^{neg}[C] := \sum_{X \in H[last(X_i)], X_i \subseteq X} P^{neg}[X]$;
11:  else
12:    $G[X_i] :=$ false;
13:  end if
14: end for
15: for each $C$ such that $R[C] \neq \{\}$ do
16:   output $R[C]$, $C$, $S^{pos}[C]$, and $S^{neg}[C]$;
17: end for
```

Proposition 4.2:
Let X be a generator. Then the closed pattern of X is $\cap$max{X’’| X’$\in$H[last(X)],X$\subseteq$ X’, X’ prefix of X’’, P[X’] $\geq$P[X’’]}.

PROOF. *Let X be a generator. Let C be the unique closed pattern of the equivalence class of X. Then C is in every transaction T that contains X. Let $X' \in H[last(X)]$ such that $X \subseteq X'$. Then C is in every transaction T that contains X'. By construction, $S = \max\{X'' \mid X'$ is prefix of X'', $P[X''] \leq P[X']\}$ are precisely those transactions having X' as a prefix. In other words, $S = f(X', \mathcal{D}) = f(X, \mathcal{D})$. Since C is a closed pattern of $[X]_{\mathcal{D}}$, it is the largest itemset that is common to all transactions in S. Then $C = \bigcap S$.* □

Correctness of GC-growth

```
Input: Dataset $\mathcal{D} = \mathcal{D}^{pos} \cup \mathcal{D}^{neg}$, and support threshold $ms$.
Output: The generators and closed patterns of $\mathcal{F}(ms, \mathcal{D})$, as well as and their support levels.
Method:
Fill in $P[\cdot]$, $P^{pos}[\cdot]$, $P^{neg}[\cdot]$, and $H[\cdot]$ by making one pass through $\mathcal{D}$; set $G[\{\}] :=$ true; initialize $R[\cdot]$ to empty.
Let $X_1$, ..., $X_n$ be the nodes in the set-enumeration tree on the possible itemsets in $\mathcal{D}$, where $X_1 <_1 X_2 <_1 \cdots <_1 X_n$.
for $i = 1, .., n$ do
    $S[X_i] := \sum_{X \in H[last(X_i)], X_i \subseteq X} P[X]$;
    if $S[X_i] \geq ms$ and $\forall \alpha \in X_i . S[X_i] < S[X_i - \{\alpha\}]$ and $\forall \alpha \in X_i . G[X_i - \{\alpha\}] =$ true then
        $G[X_i] :=$ true;
        $C := \bigcap \max\{X'' \mid X' \in H[last(X_i)], X_i \subseteq X', X'$ is prefix of $X''$, $P[X''] \leq P[X']\}$;
        $R[C] := R[C] \cup \{X_i\}$;
        $S^{pos}[C] := \sum_{X \in H[last(X_i)], X_i \subseteq X} P^{pos}[X]$;
        $S^{neg}[C] := \sum_{X \in H[last(X_i)], X_i \subseteq X} P^{neg}[X]$;
    else
        $G[X_i] :=$ false;
    end if
end for
for each $C$ such that $R[C] \neq \{\}$ do
    output $R[C]$, $C$, $S^{pos}[C]$, and $S^{neg}[C]$;
end for
```

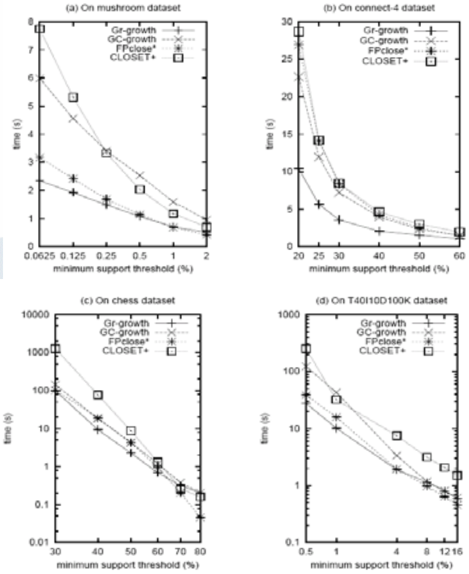
- Theorem 4.3:

  GC-growth is sound and complete for mining generators and closed patterns

---

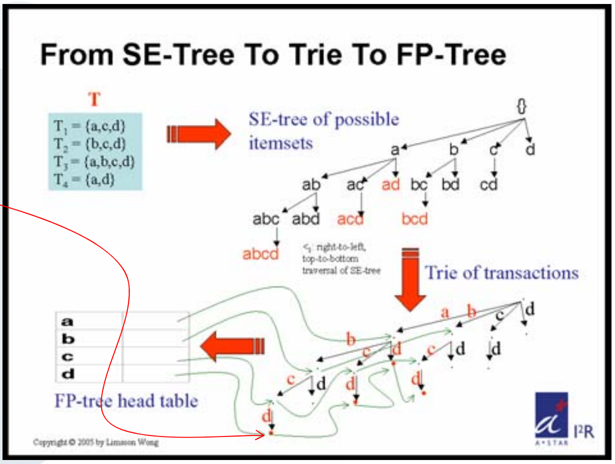
Performance of GC-growth

- GC-growth is mining both generators and closed patterns
- But is comparable in speed to the fastest algorithms that mined only closed patterns
- Also, speed of algorithms increases significantly as support threshold increases



So ...

Put transaction id's at leaves of trie so that we can get s^E o $[|\bullet|]^E$ fast



From SE-Tree To Trie To FP-Tree

T

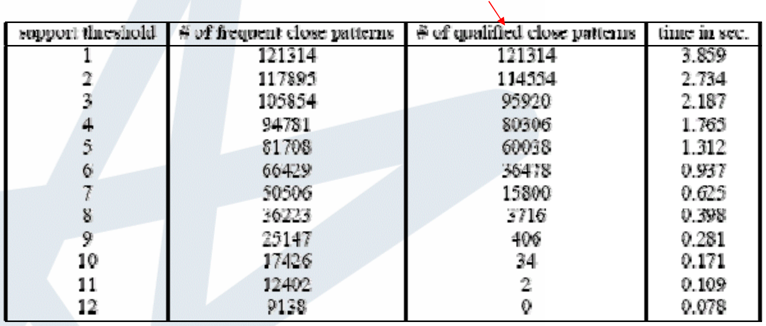
T_1 = {a,c,d}
T_2 = {b,c,d}
T_3 = {a,b,c,d}
T_4 = {a,d}

- We have an efficient algorithm for mining interesting interacting protein groups

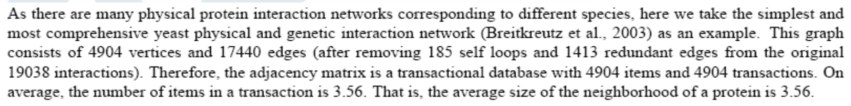
---

Example Breitkreutz et al, Genome Biology, 4, R23, 2003

X and $s^E[|X|]^E$ both occur with freq at least that of support threshold



support threshold	# of frequent close patterns	# of qualified close patterns	time in sec.
1	121314	121314	3.859
2	117895	114554	2.734
3	105854	95920	2.187
4	94781	80306	1.765
5	81708	60038	1.312
6	66429	36478	0.937
7	50506	15800	0.625
8	36223	3716	0.398
9	25147	406	0.281
10	17426	34	0.171
11	12402	2	0.109
12	9138	0	0.078



As there are many physical protein interaction networks corresponding to different species, here we take the simplest and most comprehensive yeast physical and genetic interaction network (Breitkreutz et al., 2003) as an example. This graph consists of 4904 vertices and 17440 edges (after removing 185 self loops and 1413 redundant edges from the original 19038 interactions). Therefore, the adjacency matrix is a transactional database with 4904 items and 4904 transactions. On average, the number of items in a transaction is 3.56. That is, the average size of the neighborhood of a protein is 3.56.

Generating Motif Pairs

Many Motif Discovery Methods

- MEME, Bailey & Elkan 1995
- CONSENSUS, Hertz & Stormo 1995
- PROTOMAT, Henikoff & Henikoff 1991
- CLUSTAL, Higgins & Sharp 1988
- …

- For illustration, we use PROTOMAT here

PROTOMAT

- Core of Block Maker, a WWW server that return blocks (ungapped multiple alignments) for any submitted set of protein sequences
- Comprises 2 steps:
  - MOTIF, Smith et al. 1990
    - Look for spaced triplets that occur in given set of proteins
  - MOTOMAT, Henikoff & Henikoff 1991
    - Merge overlapping candidate blocks produced by MOTIF
    - Extend blocks in both directions until similarity falls off
    - Determine best set of blocks that are in the same order and do not overlap

we treat every block, instead of whole set of blocks generated by PROTOMAT, as a binding motif

---

Example, Breitkreutz et al, Genome Biology, 4, R23, 2003

- Comprises 19038 genetic and physical interactions in yeast among 4907 proteins
- Look for interesting pairs with m = n = 5
- About 1s to generate 60k closed patterns

⇒ Too many for PROTOMAT. So consider only maximal closed patterns, giving 7847 pairs

- PROTOMAT produces 17256 left blocks and 19350 right blocks after 6 hours
- Most groups yield 1 to 3 blocks
- Ave length of blocks = 11.696, std dev = 5.45

Results & Validation

Databases Used for Validation

- BLOCKS, Pietrokovski et al. 1996
- PRINTS, Attwood & Beck 1994
- Pfam, Sonnhammer et al. 1997
- InterDom, Ng et al. 2003

	BLOCKS	PRINTS	Pfam	InterDom
Version	14.0	37.0	16.0	1.1
Num. of groups / families	4944	1850	7677	3535
Num. of entries	24294	11170	7677	30037

Validation for Single Motifs

- Compare all single motifs in our discovered motif pairs with all domains of specific domain databases
  - LAMA, Pietrokovski 1996
  - transform blocks into position-specific scoring matrices (PSSM)
  - run Smith-Waterman to align pairs of PSSM using Pearson correlation coefficient to measure similarity betw 2 columns
  - a block is mapped to another block if 95% of positions in a block occuring in the optimal alignment is common to another block and Z-score is > 5.6, where Z-score is the number std dev away from the mean generated by millions of shuffles of the BLOCKS database
- Determine number of motifs that can be mapped to these domains and the overall correlation in the portions that are mapped

---

Results for Single Motifs

	Mapped / total num. in BLOCKS	Mapped / total num. in PRINTS	Mapped / total num. in BOTH
Unique blocks	8401 / 24294	2872/ 11170	11273/ 35464
Unique groups	3568 / 4944	1325/ 1850	4893 / 6794

- Our blocks map to 32% of blocks in BLOCKS and PRINTS, yet motifs from our blocks cover 72% of domains in BLOCKS and PRINTS

⇒ Maybe most domains in BLOCKS and PRINTS have less than half a block as binding motifs, or may not be related to binding behaviour

Validation for Motif Pairs

- Map our motif pairs into domain-domain interacting pairs
- Determine the number of overlaps between our motif pairs and those in the domain-domain interaction database

- Use InterDom as the domain-domain interaction database

30037
interactions
among
3535 domains

	BLOCKS	PRINTS	Pfam	InterDom
Version	14.0	37.0	16.0	1.1
Num. of groups / families	4944	1850	7677	3535
Num. of entries	24294	11170	7677	30037

Linking Our Motif Pairs to InterDom

- InterDom represents domains by Pfam entries

⇒ To x-link, we have to
  - – Map our motifs to blocks in BLOCKS and PRINTS
  - – Link from BLOCKS and PRINTS to InterPro
  - – Link from InterPro to Pfam
  - – Match Pfam to InterDom

Results for Motif Pairs

Domain-domain interactions inferred from protein complexes or from interactions between single domain proteins

	BLOCKS overlaps	PRINTS overlaps	Combined overlaps	Confident overlaps	Complex confirmed
Domain pairs	862	26	1163	396	241

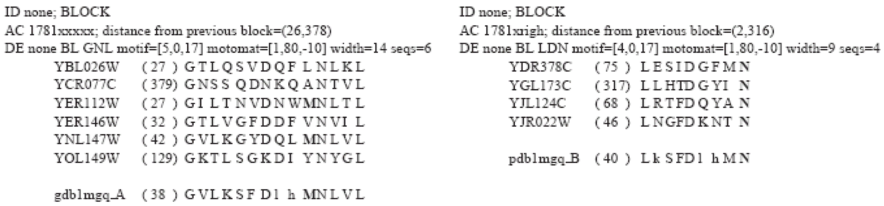
Both sides mapped to BLOCKS

Both sides mapped to PRINTS

One side mapped to PRINTS, one side mapped to BLOCKS

Example Confirmed Binding Motif

- 1 of the 241 binding motifs we found that can be confirmed using protein complexes is #1781...



```
ID none; BLOCK
AC 1781xxxxx; distance from previous block=(26,378)
DE none BL GNL motif=[5,0,17] motomat=[1,80,-10] width=14 seqs=6
        YBL026W    ( 27 ) G T L Q S V D Q F  L N L K L
        YCR077C    ( 379) G N S S  Q D N K Q  A N T V L
        YER112W    ( 27 ) G I  L T N V D N W M N L T L
        YER146W    ( 32 ) G T L V G F D D F  V N V I  L
        YNL147W    ( 42 ) G V L K G Y D Q L  M N L V L
        YOL149W    ( 129) G K T L  S G K D I   Y N Y G L

        gdb1mgq_A  ( 38 ) G V L K S F  D 1   h  M N L V L
```

```
ID none; BLOCK
AC 1781xrigh; distance from previous block=(2,316)
DE none BL LDN motif=[4,0,17] motomat=[1,80,-10] width=9 seqs=4
        YDR378C    ( 75 )  L E S I D G F M N
        YGL173C    ( 317)  L L H T D G Y I   N
        YJL124C    ( 68 )  L R T F D Q Y A  N
        YJR022W    ( 46 )  L N G F D K N T  N

        pdb1mgq_B  ( 40 )  L k  S F D 1   h M N
```

As shown in the next slide, this pair corresponds to interaction sites between LSM domains. E.g., all 7 pairs of adjacent LSM domains of pdb1mgq exhibits it.

Example: LSM Domains of pdb1mgq

Fig. 2. The structure of the complex pdb1mgq consisting of 7 LSM domains corresponding to chain A, Chain B, . . . , to chain G.

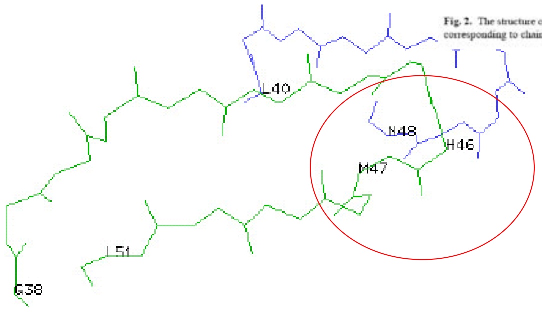


Fig. 3. Interactions between segment [38G, 51L] of LSM A and segment [40L,48N] of LSM B in the complex pdb1mgq (showing only the backbone).

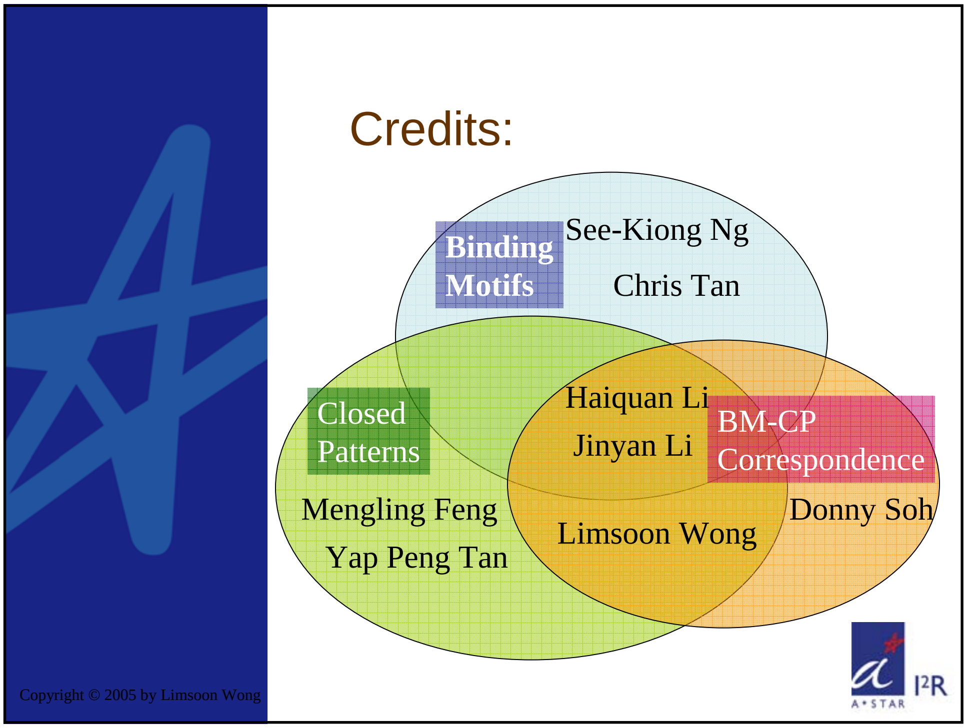
Conclusions

- Connection between maximal complete bipartite subgraphs and closed patterns

⇒ Closed pattern mining algorithms can be used to enumerate maximal complete bipartite subgraphs efficiently

- Connection between pairs of interacting protein groups and closed patterns

⇒ Discovery of binding motifs is accelerated because we need not execute expensive motif discovery algorithms on insignificant groups



Credits:

Binding Motifs

See-Kiong Ng

Chris Tan

Closed Patterns

Haiquan Li

Jinyan Li

BM-CP Correspondence

Mengling Feng

Yap Peng Tan

Limsoon Wong

Donny Soh