Smart-pooling for interactome mapping

Nicolas Thierry-Mieg
CNRS / TIMC-IMAG / TIMB, Grenoble

collaboration with Marc Vidal, CCSB / DFCI, Boston

TSB Workshop, Grenoble 10/10/2007
CCSB-HI1

Assay: yest two-hybrid (Y2H)
Space: 8100x8100
2800 interactions
125 retested by co-AP: ~80% success
-> few (technical) false positives, but many false negatives

Protocol:
• one bait against mini-pools of 188 preys, 96-well format
• identification by sequencing
• pairwise retests
Smart-pooling

Y2H and many other HT experiments:

- basic **yes-or-no test** to a large collection of “objects”
- **low-frequency** positives
- **experimental noise**

**Smart-pooling**: increase **efficiency**, **accuracy** and **coverage**, provided that

- objects individually available (eg ORFeome)
- basic assay works on pools (logical OR)
- Cherry-picking robot...

**Method:**

- **small number** of **redundant** pools
- **direct identification** (eg no sequencing in Y2H)
- deal with **false positives & negatives**
Example: rows-and-columns design

16 probes (A1-D4)
one pool per row (A-D) & column (1-4)
If C and 2 positive, then C2 is the only positive probe.
Example: rows-and-columns design

16 probes (A1-D4)
one pool per row (A-D) & column (1-4)

If **C and 2 positive**, then C2 is the only positive probe.
But if **B and 3 also positive**, the two solutions (B2 and C3) or (B3 and C2) cannot be distinguished.
Resolved by adding 4 'diagonal' pools.
Still, not a great design!

(from: Thierry-Mieg N. Pooling in systems biology becomes smart.
Nat Methods. 2006 Mar;3(3):161-2.)
The pooling problem

- Pooling problem (Combinatorial Group Testing problem) \((n, t, E)\):
  - \(\mathcal{A}_n\), a set of Boolean variables \((n \approx 100-10^4)\)
  - \(t = \text{max number of positives} (\approx 1-10)\)
  - \(E = \text{max number of errors} (\approx 1-40\% \text{ of tests})\)

Pool: subset of \(\mathcal{A}_n\), value=OR

Goal: build a set of \(v\) pools
- \(v\) as small as possible
- guarantee correction of errors & identification of positives
Matrix representation

A boolean matrix $M(i,j)$ is true if pool $i$ contains variable $j$.

Example: $n=9$, $A_9 = \{0, 1, \ldots, 8\}$:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Pools:
- $\{0, 3, 6\}$
- $\{1, 4, 7\}$
- $\{2, 5, 8\}$

“Layer” = partition of $A_n$
Shifted Transversal Design: idea

“Transversal” construction: layers
“Shift” variables from layer to layer
- Limit co-occurrence of variables
- Constant-sized intersections between pools
Shifted Transversal Design: idea

“Transversal” construction: layers
“Shift” variables from layer to layer
- Limit co-occurrence of variables
- Constant-sized intersections between pools

STD(n;q;k) : n variables, q prime, q < n, k number of layers (k ≤ q+1)
- First q layers: symmetric construction, q pools of size n/q or 1+n/q
- If k=q+1 : additional singular layer, up to q pools of heterogeneous sizes
Shifted Transversal Design: idea

“Transversal” construction: layers
“Shift” variables from layer to layer
- Limit co-occurrence of variables
- Constant-sized intersections between pools

STD(n;q;k) : n variables, q prime, q < n, k number of layers (k ≤ q+1)
- First q layers: symmetric construction, q pools of size n/q or 1+n/q
- If k=q+1 : additional singular layer, up to q pools of heterogeneous sizes

Let:
- $\sigma_q$ circular permutation on $\{0,1\}^q$ : $\sigma_q \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_q \end{bmatrix} = \begin{bmatrix} x_q \\ x_1 \\ \vdots \\ x_{q-1} \end{bmatrix}$
- $\Gamma(q,n) = \min\{\gamma \mid q^{\gamma+1} \geq n\}$
\[ \forall j \in \{0,\ldots,q\}: M_j \text{ qxn Boolean matrix, representing layer } L(j) \]

columns \[ C_{j,0}, \ldots, C_{j,n-1} : \]

\[ C_{0,0} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} , \text{ and } \forall i \in \{0,\ldots,n\} \ C_{j,i} = \sigma^s(i,j)(C_{0,0}) \text{ where:} \]

- if \( j < q \): \[ s(i,j) = \sum_{c=0}^{I} j^c \left\lfloor \frac{i}{q^c} \right\rfloor \]
- If \( j = q \) (singular layer): \[ s(i,q) = \left\lfloor \frac{i}{q^I} \right\rfloor \]

For \( k \in \{1,\ldots,q+1\} \), \( \text{STD}(n;q;k) = L(0) \cup \ldots \cup L(k-1) \)
STD example: $n=9$, $q=3$

$L(0) = \{\{0,3,6\},\{1,4,7\},\{2,5,8\}\}$

$L(1) = \{\{0,5,7\},\{1,3,8\},\{2,4,6\}\}$

$L(2) = \{\{0,4,8\},\{1,5,6\},\{2,3,7\}\}$

$L(3) = \{\{0,1,2\},\{3,4,5\},\{6,7,8\}\}$

STD($n=9;q=3;k=2$) = $L(0) \cup L(1)$
STD example: n=9 to 27, q=3

n=9, q=3, third layer (j=2): \[ M_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \]

n=27, q=3, j=2: \[ s(i, j) = \sum_{c=0}^{T} j^c \left[ \frac{i}{q^c} \right] = i + 2 \left[ \frac{i}{3} \right] + 4 \left[ \frac{i}{9} \right] + (1+j+j^2) \]

\[
M_2 = \begin{bmatrix}
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\]
STD Properties

- Theorem: number of pools that contain any 2 variables is at most \( \Gamma(q,n) \)

- Proof: layers \( j = \) roots of non-zero polynomial on GF(q) of degree at most \( \Gamma \)

- Example: \( n=9, q=3 \)
  
  \[
  L(0) = \{\{0,3,6\}, \{1,4,7\}, \{2,5,8\}\}
  \]
  
  \[
  L(1) = \{\{0,5,7\}, \{1,3,8\}, \{2,4,6\}\}
  \]
  
  \[
  L(2) = \{\{0,4,8\}, \{1,5,6\}, \{2,3,7\}\}
  \]
  
  \[
  L(3) = \{\{0,1,2\}, \{3,4,5\}, \{6,7,8\}\}
  \]

  0 appears exactly once (\( \Gamma=1 \)) with each other variable.
A solution to the pooling problem

- **Corollary:** If there are at most $t$ positive variables in $\mathcal{A}_n$ and at most $E$ false positive and $E$ false negative observations: STD($n;q;k$) is a solution, when choosing $q$ prime such that $t \cdot \Gamma(q,n) + 2 \cdot E \leq q$, and $k = t \cdot \Gamma + 2 \cdot E + 1$

- Constructive proof: exhibit a simple algorithm that works
  Algorithm relies on knowledge of $E$
A solution to the pooling problem

- **Corollary:** If there are at most \( t \) positive variables in \( \mathcal{A}_n \) and at most \( E \) false positive and \( E \) false negative observations: \( \text{STD}(n;q;k) \) is a solution, when choosing \( q \) prime such that \( t \cdot \Gamma(q,n)+2 \cdot E \leq q \), and \( k = t \cdot \Gamma+2 \cdot E+1 \)

- Constructive proof: exhibit a simple algorithm that works
  Algorithm relies on knowledge of \( E \)

- STD is sound

- Allows to compare with other published designs: favorable (on numerical examples)
Even redistribution of variables

**Theorem:** Let $m \leq k \leq q$ and consider $\{P_1, \ldots, P_m\} \subset \text{STD}(n;q;k)$, each belonging to a different layer. Then:

$$\lambda_m \leq \left| \bigcap_{h=1}^{m} P_h \right| \leq \lambda_m + 1,$$

where

$$\lambda_m = \sum_{c=m}^{r} \left\lfloor \frac{n-1}{q^c} \right\rfloor \% q \cdot q^{c-m}.$$

**Notes:**

- $\lambda_m$ depends only on $m$, not on the choice of the pools $P_1, \ldots, P_m$
- every pool, and every intersection between 2 or more pools, is redistributed evenly in each remaining layer
- $L(q)$ does not work ($k \leq q$)
In practice: tolerate a few ambiguous variables → many fewer pools
Example: n=10000, t=5, error-rate 1%
- guarantee requires 483 pools
- when tolerating up to 10 ambiguous variables, 143 pools prove sufficient

Given (n,t,E-rates) and “ambiguity tolerance”, find optimal parameter values by simulation

Difficulty: “decode” observed pool values
Interpreting smart-pooling results

Decoding an observation: a combinatorial optimization problem

Difficult for general solvers (e.g., integer linear programming)

Interpool: an algorithm to solve it
  - Branch-and-bound
  - Exact
  - Fast (usually)
  - GNU GPL

Manuscript under review
Validation

- Pilot project: 100 baits x 940 preys
- Varied subspace of CCSB-HI1: many interactions, hubs, auto-activators...

Choosing the design: simulations with interpools
### STD(940;13;13), 10% FPR

<table>
<thead>
<tr>
<th>Positives</th>
<th>FNR</th>
<th>TPs missed</th>
<th>Retests</th>
<th>Simulations</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>10%</td>
<td>0</td>
<td>2.26</td>
<td>10000</td>
<td>1m</td>
</tr>
<tr>
<td></td>
<td>20%</td>
<td>0</td>
<td>2.26</td>
<td></td>
<td>1m</td>
</tr>
<tr>
<td></td>
<td>30%</td>
<td>1.2%</td>
<td>2.27</td>
<td></td>
<td>4m</td>
</tr>
<tr>
<td>3</td>
<td>10%</td>
<td>0</td>
<td>3.57</td>
<td>10000</td>
<td>4m</td>
</tr>
<tr>
<td></td>
<td>20%</td>
<td>0.4%</td>
<td>3.58</td>
<td></td>
<td>33m</td>
</tr>
<tr>
<td></td>
<td>30%</td>
<td>3.4%</td>
<td>3.60</td>
<td></td>
<td>2h</td>
</tr>
<tr>
<td>4</td>
<td>10%</td>
<td>0</td>
<td>5.06</td>
<td>10000</td>
<td>32m</td>
</tr>
<tr>
<td></td>
<td>20%</td>
<td>1.0%</td>
<td>5.11</td>
<td>10000</td>
<td>10h39m</td>
</tr>
<tr>
<td></td>
<td>30%</td>
<td>6.2%</td>
<td>5.26</td>
<td>7500</td>
<td>2d11h</td>
</tr>
<tr>
<td>5</td>
<td>10%</td>
<td>0.1%</td>
<td>6.71</td>
<td>10000</td>
<td>4h</td>
</tr>
<tr>
<td></td>
<td>20%</td>
<td>1.7%</td>
<td>6.94</td>
<td>1000</td>
<td>12h47m</td>
</tr>
<tr>
<td></td>
<td>30%</td>
<td>12.9%</td>
<td>7.88</td>
<td>300</td>
<td>3d10h</td>
</tr>
</tbody>
</table>

TPs missed and Retests: upper bounds of the 95% confidence intervals
Validation

- Pilot project: 100 baits x 940 preys
- Varied subspace of CCSB-HI1: many interactions, hubs, auto-activators...
- Smart-pooled the 940 preys according to STD(940;13;13)
  - 169 pools, 73 preys in each pool
  - each prey is in 13 pools
  - at most 2 pools contain any pair
    → 3 pools for identification, 10 pools for errors and multiple positives
- Screened each bait against the 169 pools, scored positive pools
- Decoded the patterns of positive pools (interpool) -> putative positives
- Pairwise retests
Circles: spots scored positive.
Decoding finds:
- 2 interactors: green (no FNs), and blue (3 FNs = red arrows)
- 2 FPs (red circles)
Results

- Identified 65 putative interactions

- Retest: 60 passed, 3 failed, 2 unconfirmed (auto-activation in the retest)
  → Specificity between 92% and 95%
Results

- Identified 65 putative interactions

- Retest: 60 passed, 3 failed, 2 unconfirmed (auto-activation in the retest)
  → **Specificity between 92% and 95%**

- 60 confirmed = 36 CCSB-HI1 + 24 novel

  - **Recall of CCSB-HI1 data**: the 36 represent 73% of CCSB-HI1, or 84% when excluding the two hardest baits (strong hub, auto-activator)
  - **Sensitivity vs CCSB-HI1**: Difficult because subspace strongly biased
  
  Low estimate: **172% higher sensitivity**
  
  High estimate: **325% higher sensitivity**
STD (the Shifted Transversal Design) is a **flexible and efficient family of pooling designs.** On paper and in silico, STD performs very well.

Interpool is a **fast exact decoding algorithm.** Useful both for choosing a design (simulations) and for interpreting experimental results. Open source.

Smart-pooling really works for HT-Y2H: it is efficient, sensitive and specific.
**Current work:** scaling up to the complete *C. elegans* ORFeome, using denser formats (384 and 1536)

Takes advantage of STD symmetries: build micro-pools, then combine at will
Acknowledgments

M. Vidal, D. Hill, J.-F. Rual: CCSB, Dana-Farber Cancer Institute, Boston

C. Boone, X. Xin: Boone Lab, University of Toronto

J.-L. Roch, L. Trilling, G. Bailly: TIMC-IMAG and LIG, Grenoble

Funding: INPG, Grenoble (2003-2004)
