Technische Universiteit
Eindhoven
University of Technology

Efficient Probabilistic Group Testing Based on Traitor Tracing

Thijs Laarhoven

mail@thijs.com
http://www.thijs.com/

Monticello, Illinois, USA
(October 4, 2013)

Problem: Blood Testing

	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪
Amin												
Boris												
Caroline												
David												
Eve												
Fred												
Gábor												
Henry												

Problem: Blood Testing

	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪
Amin	1	0	0	0	0	0	0	0				
Boris	0	1	0	0	0	0	0	0				
Caroline	0	0	1	0	0	0	0	0				
David	0	0	0	1	0	0	0	0				
Eve	0	0	0	0	1	0	0	0				
Fred	0	0	0	0	0	1	0	0				
Gábor	0	0	0	0	0	0	1	0				
Henry	0	0	0	0	0	0	0	1				

Problem: Blood Testing

	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪
Amin	1	0	0	0	0	0	0	0				
Boris	0	1	0	0	0	0	0	0				
Caroline	0	0	1	0	0	0	0	0				
David	0	0	0	1	0	0	0	0				
Eve	0	0	0	0	1	0	0	0				
Fred	0	0	0	0	0	1	0	0				
Gábor	0	0	0	0	0	0	1	0				
Henry	0	0	0	0	0	0	0	1				
Results	0	0	1	0	0	0	0	0				

Problem: Blood Testing

	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪
Amin	1	0	0	0	0	0	0	0				
Boris	0	1	0	0	0	0	0	0				
Caroline	0	0	1	0	0	0	0	0				
David	0	0	0	1	0	0	0	0				
Eve	0	0	0	0	1	0	0	0				
Fred	0	0	0	0	0	1	0	0				
Gábor	0	0	0	0	0	0	1	0				
Henry	0	0	0	0	0	0	0	1				
Results	0	0	1	0	0	0	0	0				

Solution: Using Pools

	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪
Amin	0	0	0									
Boris	0	0	1									
Caroline	0	1	0									
David	0	1	1									
Eve	1	0	0									
Fred	1	0	1									
Gábor	1	1	0									
Henry	1	1	1									
Results												

Solution: Using Pools

	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪
Amin	0	0	0									
Boris	0	0	1									
Caroline	0	1	0									
David	0	1	1									
Eve	1	0	0									
Fred	1	0	1									
Gábor	1	1	0									
Henry	1	1	1									
Results	0	1	0									

Solution: Using Pools

	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪
Amin	0	0	0									
Boris	0	0	1									
Caroline	0	1	0									
David	0	1	1									
Eve	1	0	0									
Fred	1	0	1									
Gábor	1	1	0									
Henry	1	1	1									
Results	0	1	0									

Problem: Multiple (K) Infected

	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪
Amin	0	0	0									
Boris	0	0	1									
Caroline	0	1	0									
David	0	1	1									
Eve	1	0	0									
Fred	1	0	1									
Gábor	1	1	0									
Henry	1	1	1									
Results												

Problem: Multiple (K) Infected

	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪
Amin	0	0	0									
Boris	0	0	1									
Caroline	0	1	0									
David	0	1	1									
Eve	1	0	0									
Fred	1	0	1									
Gábor	1	1	0									
Henry	1	1	1									
Results												

Problem: Multiple (K) Infected

	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪
Amin	0	0	0									
Boris	0	0	1									
Caroline	0	1	0									
David	0	1	1									
Eve	1	0	0									
Fred	1	0	1									
Gábor	1	1	0									
Henry	1	1	1									
Results	1	1	1									

Solution: Group Testing

	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪
Amin	?	?	?	?	?	?	?	?	?	?	?	?
Boris	?	?	?	?	?	?	?	?	?	?	?	?
Caroline	?	?	?	?	?	?	?	?	?	?	?	?
David	?	?	?	?	?	?	?	?	?	?	?	?
Eve	?	?	?	?	?	?	?	?	?	?	?	?
Fred	?	?	?	?	?	?	?	?	?	?	?	?
Gábor	?	?	?	?	?	?	?	?	?	?	?	?
Henry	?	?	?	?	?	?	?	?	?	?	?	?
Results	?	?	?	?	?	?	?	?	?	?	?	?

Solution: Group Testing

	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪
Amin	?	?	?	?	?	?	?	?	?	?	?	?
Boris	?	?	?	?	?	?	?	?	?	?	?	?
Caroline	?	?	?	?	?	?	?	?	?	?	?	?
David	?	?	?	?	?	?	?	?	?	?	?	?
Eve	?	?	?	?	?	?	?	?	?	?	?	?
Fred	?	?	?	?	?	?	?	?	?	?	?	?
Gábor	?	?	?	?	?	?	?	?	?	?	?	?
Henry	?	?	?	?	?	?	?	?	?	?	?	?
Results	?	?	?	?	?	?	?	?	?	?	?	?

1. An algorithm to construct group testing matrices

Solution: Group Testing

	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪
Amin	?	?	?	?	?	?	?	?	?	?	?	?
Boris	?	?	?	?	?	?	?	?	?	?	?	?
Caroline	?	?	?	?	?	?	?	?	?	?	?	?
David	?	?	?	?	?	?	?	?	?	?	?	?
Eve	?	?	?	?	?	?	?	?	?	?	?	?
Fred	?	?	?	?	?	?	?	?	?	?	?	?
Gábor	?	?	?	?	?	?	?	?	?	?	?	?
Henry	?	?	?	?	?	?	?	?	?	?	?	?
Results	?	?	?	?	?	?	?	?	?	?	?	?

1. An algorithm to construct group testing matrices
2. An algorithm to link test results to infected people

Solution: Group Testing

	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪	🧪
Amin	$X \in \{0,1\}^{N \times T}$											
Boris												
Caroline												
David												
Eve												
Fred												
Gábor												
Henry												
Results	$y \in \{0,1\}^{T}$											

1. An algorithm to construct group testing matrices
2. An algorithm to link test results to infected people

Solution: Group Testing

	🧪 🧪 🧪 🧪 🧪 🧪 🧪 🧪 🧪 🧪 🧪 🧪
Amin	
Boris	$X \in \{0,1\}^{N\times T}$
Caroline	
David	
Eve	
Fred	deterministic: $T = \Omega(K^2 \log N)$
Gábor	
Henry	
Results	$y \in \{0,1\}^T$

1. An algorithm to construct group testing matrices
2. An algorithm to link test results to infected people

Solution: Group Testing

Amin Boris Caroline David Eve Fred Gábor Henry	$X \in \{0,1\}^{N \times T}$ deterministic: $T = \Omega(K^2 \log N)$ probabilistic: $T = \Theta(K \log N)$
Results	$y \in \{0,1\}^T$

1. An algorithm to construct group testing matrices
2. An algorithm to link test results to infected people

TU/e

Solution: Group Testing

Amin Boris Caroline David Eve Fred Gábor Henry	$X \in \{0,1\}^{N \times T}$ deterministic: $T = \Omega(K^2 \log N)$ probabilistic: $T = \Theta(K \log N)$
Results	$y \in \{0,1\}^T$

1. An algorithm to construct group testing matrices
2. An algorithm to link test results to infected people

Solution: Group Testing

1. An algorithm to construct group testing matrices
2. An algorithm to link test results to infected people

Solution: Group Testing

1. An algorithm to construct group testing matrices

2. An algorithm to link test results to infected people

A Group Testing Framework

1. An algorithm to construct group testing matrices

2. An algorithm to link test results to infected people

A Group Testing Framework

1. An algorithm to construct group testing matrices
   1a. For each test i, person j, choose $X_{j,i} = 1$ with prob. p.
      - Intuitively: $p \approx \frac{1}{K}$.
      - Precise value of p depends on N, K, ε.
2. An algorithm to link test results to infected people

A Group Testing Framework

1. An algorithm to construct group testing matrices

   1a. For each test i, person j, choose $X_{j,i} = 1$ with prob. p.

   - Intuitively: $p \approx \frac{1}{K}$.
   - Precise value of p depends on N, K, ε.

2. An algorithm to link test results to infected people

   2a. For each test i, person j, compute $S_{j,i} = g(X_{j,i}, y_i)$.

   - Positive scores ($S_{j,i} > 0$) for matches ($X_{j,i} = y_i$).
   - Negative scores ($S_{j,i} < 0$) for differences ($X_{j,i} \neq y_i$).
   - Large scores ($|S_{j,i}| \gg 0$) for rare events.

A Group Testing Framework

1. An algorithm to construct group testing matrices

   1a. For each test i, person j, choose $X_{j,i} = 1$ with prob. p.

   - Intuitively: $p \approx \frac{1}{K}$.
   - Precise value of p depends on N, K, ε.

2. An algorithm to link test results to infected people

   2a. For each test i, person j, compute $S_{j,i} = g(X_{j,i}, y_i)$.

   - Positive scores ($S_{j,i} > 0$) for matches ($X_{j,i} = y_i$).
   - Negative scores ($S_{j,i} < 0$) for differences ($X_{j,i} \neq y_i$).
   - Large scores ($|S_{j,i}| \gg 0$) for rare events.

   2b. Mark person j infected iff $\sum_i S_{j,i} > Z$ (threshold).

   - Large Z: Fewer false positives, more false negatives.
   - Small Z: More false positives, fewer false negatives.

A Group Testing Framework

1. An algorithm to construct group testing matrices
   - 1a. For each test i, person j, choose $X_{j,i} = 1$ with prob. p.
     - Intuitively: $p \approx \frac{1}{K}$.
     - Precise value of p depends on N, K, ε.
2. An algorithm to link test results to infected people
   - 2a. For each test i, person j, compute $S_{j,i} = g(X_{j,i}, y_i)$.
     - Positive scores ($S_{j,i} > 0$) for matches ($X_{j,i} = y_i$).
     - Negative scores ($S_{j,i} < 0$) for differences ($X_{j,i} \neq y_i$).
     - Large scores ($|S_{j,i}| \gg 0$) for rare events.
   - 2b. Mark person j infected iff $\sum_i S_{j,i} > Z$ (threshold).
     - Large Z: Fewer false positives, more false negatives.
     - Small Z: More false positives, fewer false negatives.

Exact choices of p, g, and Z depend on the model/parameters.

Traditional Group Testing

1. An algorithm to construct group testing matrices
   1a. For each test i, person j, choose $X_{j,i} = 1$ with prob. p.

2. An algorithm to link test results to infected people
   2a. For each test i, person j, compute $S_{j,i} = g(X_{j,i}, y_i)$.

   2b. Mark person j infected iff $\sum_i S_{j,i} > Z$ (threshold).

Traditional Group Testing

1. An algorithm to construct group testing matrices
   1a. For each test i, person j, choose $X_{j,i} = 1$ with prob. p.

2. An algorithm to link test results to infected people
   2a. For each test i, person j, compute $S_{j,i} = g(X_{j,i}, y_i)$.

$$g(X_{j,i}, y_i) = \begin{cases} +p/(1-p), & \text{if } X_{j,i} = 0, y_i = 0, \\ -p(1-p)^{K-1}/(1-(1-p)^K), & \text{if } X_{j,i} = 0, y_i = 1, \\ -1, & \text{if } X_{j,i} = 1, y_i = 0, \\ +(1-p)^K/(1-(1-p)^K), & \text{if } X_{j,i} = 1, y_i = 1. \end{cases}$$

   2b. Mark person j infected iff $\sum_i S_{j,i} > Z$ (threshold).

Traditional Group Testing

1. An algorithm to construct group testing matrices
   1a. For each test i, person j, choose $X_{j,i} = 1$ with prob. p.

$$p = \mathrm{argmin}_q T(N, K, \varepsilon, q).$$

2. An algorithm to link test results to infected people
   2a. For each test i, person j, compute $S_{j,i} = g(X_{j,i}, y_i)$.

$$g(X_{j,i}, y_i) = \begin{cases} +p/(1-p), & \text{if } X_{j,i} = 0, y_i = 0, \\ -p(1-p)^{K-1}/(1-(1-p)^K), & \text{if } X_{j,i} = 0, y_i = 1, \\ -1, & \text{if } X_{j,i} = 1, y_i = 0, \\ +(1-p)^K/(1-(1-p)^K), & \text{if } X_{j,i} = 1, y_i = 1. \end{cases}$$

   2b. Mark person j infected iff $\sum_i S_{j,i} > Z$ (threshold).

Traditional Group Testing

1. An algorithm to construct group testing matrices

   1a. For each test i, person j, choose $X_{j,i} = 1$ with prob. p.

$$p = \mathrm{argmin}_q T(N, K, \varepsilon, q).$$

2. An algorithm to link test results to infected people

   2a. For each test i, person j, compute $S_{j,i} = g(X_{j,i}, y_i)$.

$$g(X_{j,i}, y_i) = \begin{cases} +p/(1-p), & \text{if } X_{j,i} = 0, y_i = 0, \\ -p(1-p)^{K-1}/(1-(1-p)^K), & \text{if } X_{j,i} = 0, y_i = 1, \\ -1, & \text{if } X_{j,i} = 1, y_i = 0, \\ +(1-p)^K/(1-(1-p)^K), & \text{if } X_{j,i} = 1, y_i = 1. \end{cases}$$

   2b. Mark person j infected iff $\sum_i S_{j,i} > Z$ (threshold).

$$Z = Z(N, K, \varepsilon, p),$$
$$T = T(N, K, \varepsilon, p).$$

TU/e

Example: $N = 8$, $K = 3$, $\varepsilon = 10^{-2}$

1. An algorithm to construct group testing matrices
   1a. For each test i, person j, choose $X_{j,i} = 1$ with prob. p.

$$p = \mathrm{argmin}_q T(N, K, \varepsilon, q).$$

2. An algorithm to link test results to infected people
   2a. For each test i, person j, compute $S_{j,i} = g(X_{j,i}, y_i)$.

$$g(X_{j,i}, y_i) = \begin{cases} +p/(1-p), & \text{if } X_{j,i} = 0, y_i = 0, \\ -p(1-p)^{K-1}/(1-(1-p)^K), & \text{if } X_{j,i} = 0, y_i = 1, \\ -1, & \text{if } X_{j,i} = 1, y_i = 0, \\ +(1-p)^K/(1-(1-p)^K), & \text{if } X_{j,i} = 1, y_i = 1. \end{cases}$$

   2b. Mark person j infected iff $\sum_i S_{j,i} > Z$ (threshold).

$$Z = Z(N, K, \varepsilon, p),$$
$$T = T(N, K, \varepsilon, p).$$

Example: $N = 8$, $K = 3$, $\varepsilon = 10^{-2}$

1. An algorithm to construct group testing matrices
   1a. For each test i, person j, choose $X_{j,i} = 1$ with prob. p.

$$p = 0.25\dots.$$

2. An algorithm to link test results to infected people
   2a. For each test i, person j, compute $S_{j,i} = g(X_{j,i}, y_i)$.

$$g(X_{j,i}, y_i) = \begin{cases} +p/(1-p), & \text{if } X_{j,i} = 0, y_i = 0, \\ -p(1-p)^{K-1}/(1-(1-p)^K), & \text{if } X_{j,i} = 0, y_i = 1, \\ -1, & \text{if } X_{j,i} = 1, y_i = 0, \\ +(1-p)^K/(1-(1-p)^K), & \text{if } X_{j,i} = 1, y_i = 1. \end{cases}$$

   2b. Mark person j infected iff $\sum_i S_{j,i} > Z$ (threshold).

$$Z = Z(N, K, \varepsilon, p),$$
$$T = T(N, K, \varepsilon, p).$$

Example: $N = 8$, $K = 3$, $\varepsilon = 10^{-2}$

1. An algorithm to construct group testing matrices
   1a. For each test i, person j, choose $X_{j,i} = 1$ with prob. p.

$$p = 0.25\ldots.$$

2. An algorithm to link test results to infected people
   2a. For each test i, person j, compute $S_{j,i} = g(X_{j,i}, y_i)$.

$$g(X_{j,i}, y_i) = \begin{cases} +0.34\ldots, & \text{if } X_{j,i} = 0, y_i = 0, \\ -0.24\ldots, & \text{if } X_{j,i} = 0, y_i = 1, \\ -1, & \text{if } X_{j,i} = 1, y_i = 0, \\ +0.69\ldots, & \text{if } X_{j,i} = 1, y_i = 1. \end{cases}$$

   2b. Mark person j infected iff $\sum_i S_{j,i} > Z$ (threshold).

$$Z = Z(N, K, \varepsilon, p),$$
$$T = T(N, K, \varepsilon, p).$$

Example: $N = 8$, $K = 3$, $\varepsilon = 10^{-2}$

1. An algorithm to construct group testing matrices

   1a. For each test i, person j, choose $X_{j,i} = 1$ with prob. p.

$$p = 0.25\dots.$$

2. An algorithm to link test results to infected people

   2a. For each test i, person j, compute $S_{j,i} = g(X_{j,i}, y_i)$.

$$g(X_{j,i}, y_i) = \begin{cases} +0.34\dots, & \text{if } X_{j,i} = 0, y_i = 0, \\ -0.24\dots, & \text{if } X_{j,i} = 0, y_i = 1, \\ -1, & \text{if } X_{j,i} = 1, y_i = 0, \\ +0.69\dots, & \text{if } X_{j,i} = 1, y_i = 1. \end{cases}$$

   2b. Mark person j infected iff $\sum_i S_{j,i} > Z$ (threshold).

$$Z = 22.62\dots,$$

$$T = 160.$$

Example: Group testing matrix

	🧪	🧪	🧪	🧪	🧪	. . .	🧪
Amin	$X_{1,1}$	$X_{1,2}$	$X_{1,3}$	$X_{1,4}$	$X_{1,5}$	$\cdots$	$X_{1,160}$
Boris	$X_{2,1}$	$X_{2,2}$	$X_{2,3}$	$X_{2,4}$	$X_{2,5}$	$\cdots$	$X_{2,160}$
Caroline	$X_{3,1}$	$X_{3,2}$	$X_{3,3}$	$X_{3,4}$	$X_{3,5}$	$\cdots$	$X_{3,160}$
David	$X_{4,1}$	$X_{4,2}$	$X_{4,3}$	$X_{4,4}$	$X_{4,5}$	$\cdots$	$X_{4,160}$
Eve	$X_{5,1}$	$X_{5,2}$	$X_{5,3}$	$X_{5,4}$	$X_{5,5}$	$\cdots$	$X_{5,160}$
Fred	$X_{6,1}$	$X_{6,2}$	$X_{6,3}$	$X_{6,4}$	$X_{6,5}$	$\cdots$	$X_{6,160}$
Gábor	$X_{7,1}$	$X_{7,2}$	$X_{7,3}$	$X_{7,4}$	$X_{7,5}$	$\cdots$	$X_{7,160}$
Henry	$X_{8,1}$	$X_{8,2}$	$X_{8,3}$	$X_{8,4}$	$X_{8,5}$	$\cdots$	$X_{8,160}$
Copy	y_1	y_2	y_3	y_4	y_5	$\cdots$	y_{160}

Example: Group testing matrix

1a. For each test i, person j, set $X_{j,i} = 1$ with prob. p.

	🧪	🧪	🧪	🧪	🧪	...	🧪
Amin	$X_{1,1}$	$X_{1,2}$	$X_{1,3}$	$X_{1,4}$	$X_{1,5}$	$\dots$	$X_{1,160}$
Boris	$X_{2,1}$	$X_{2,2}$	$X_{2,3}$	$X_{2,4}$	$X_{2,5}$	$\dots$	$X_{2,160}$
Caroline	$X_{3,1}$	$X_{3,2}$	$X_{3,3}$	$X_{3,4}$	$X_{3,5}$	$\dots$	$X_{3,160}$
David	$X_{4,1}$	$X_{4,2}$	$X_{4,3}$	$X_{4,4}$	$X_{4,5}$	$\dots$	$X_{4,160}$
Eve	$X_{5,1}$	$X_{5,2}$	$X_{5,3}$	$X_{5,4}$	$X_{5,5}$	$\dots$	$X_{5,160}$
Fred	$X_{6,1}$	$X_{6,2}$	$X_{6,3}$	$X_{6,4}$	$X_{6,5}$	$\dots$	$X_{6,160}$
Gábor	$X_{7,1}$	$X_{7,2}$	$X_{7,3}$	$X_{7,4}$	$X_{7,5}$	$\dots$	$X_{7,160}$
Henry	$X_{8,1}$	$X_{8,2}$	$X_{8,3}$	$X_{8,4}$	$X_{8,5}$	$\dots$	$X_{8,160}$
Copy	y_1	y_2	y_3	y_4	y_5	$\dots$	y_{160}

Example: Group testing matrix

1a. For each test i, person j, set $X_{j,i} = 1$ with prob. p.

	🧪	🧪	🧪	🧪	🧪	...	🧪
Amin	0	0	0	0	0	...	0
Boris	1	0	1	1	1	...	1
Caroline	0	0	0	1	0	...	0
David	0	0	1	1	1	...	0
Eve	0	0	0	0	0	...	0
Fred	1	0	1	0	0	...	0
Gábor	0	0	1	0	0	...	0
Henry	0	0	0	0	1	...	0
Copy	y_1	y_2	y_3	y_4	y_5	...	y_{160}

Example: Running the tests

Infected samples determine the test results.

	🧪	🧪	🧪	🧪	🧪	...	🧪
Amin	.	.	.	.	.	...	.
Boris	.	.	.	.	.	...	.
Caroline	0	0	0	1	0	...	0
David	.	.	.	.	.	...	.
Eve	0	0	0	0	0	...	0
Fred	.	.	.	.	.	...	.
Gábor	.	.	.	.	.	...	.
Henry	0	0	0	0	1	...	0
Results	y_1	y_2	y_3	y_4	y_5	...	y_{160}

$$\text{Infected} = \{\text{Caroline}, \text{Eve}, \text{Henry}\}$$

Example: Running the tests

Infected samples determine the test results.

	🧪	🧪	🧪	🧪	🧪	...	🧪
Amin	.	.	.	.	.	...	.
Boris	.	.	.	.	.	...	.
Caroline	0	0	0	1	0	...	0
David	.	.	.	.	.	...	.
Eve	0	0	0	0	0	...	0
Fred	.	.	.	.	.	...	.
Gábor	.	.	.	.	.	...	.
Henry	0	0	0	0	1	...	0
Results	0	0	0	1	1	...	0

$$\text{Infected} = \{\text{Caroline}, \text{Eve}, \text{Henry}\}$$

Example: Scores

We perform the tests and the results come back.

	🧪	🧪	🧪	🧪	🧪	...	🧪
Amin	0	0	0	0	0	...	0
Boris	1	0	1	1	1	...	1
Caroline	0	0	0	1	0	...	0
David	0	0	1	1	1	...	0
Eve	0	0	0	0	0	...	0
Fred	1	0	1	0	0	...	0
Gábor	0	0	1	0	0	...	0
Henry	0	0	0	0	1	...	0
Results	0	0	0	1	1	...	0

$$\text{Infected} = \{\text{Caroline}, \text{Eve}, \text{Henry}\}$$

Example: Scores

2a. For each test i, person j, compute $S_{j,i} = g(X_{j,i}, y_i)$.

	🧪	🧪	🧪	🧪	🧪	...	🧪
Amin	0	0	0	0	0	...	0
Boris	1	0	1	1	1	...	1
Caroline	0	0	0	1	0	...	0
David	0	0	1	1	1	...	0
Eve	0	0	0	0	0	...	0
Fred	1	0	1	0	0	...	0
Gábor	0	0	1	0	0	...	0
Henry	0	0	0	0	1	...	0
Results	0	0	0	1	1	...	0

$$\text{Infected} = \{\text{Caroline}, \text{Eve}, \text{Henry}\}$$

Example: Scores

2a. For each test i, person j, compute $S_{j,i} = g(X_{j,i}, y_i)$.

	🧪	🧪	🧪	🧪	🧪	...	🧪
Amin	$+0.3$	$+0.3$	$+0.3$	-0.2	-0.2	...	$+0.3$
Boris	-1.0	$+0.3$	-1.0	$+0.7$	$+0.7$	...	-1.0
Caroline	$+0.3$	$+0.3$	$+0.3$	$+0.7$	-0.2	...	$+0.3$
David	$+0.3$	$+0.3$	-1.0	$+0.7$	$+0.7$	...	$+0.3$
Eve	$+0.3$	$+0.3$	$+0.3$	-0.2	-0.2	...	$+0.3$
Fred	-1.0	$+0.3$	-1.0	-0.2	-0.2	...	$+0.3$
Gábor	$+0.3$	$+0.3$	-1.0	-0.2	-0.2	...	$+0.3$
Henry	$+0.3$	$+0.3$	$+0.3$	-0.2	$+0.7$	...	$+0.3$
Results	0	0	0	1	1	...	0

$$\text{Infected} = \{\text{Caroline}, \text{Eve}, \text{Henry}\}$$

Example: Scores

2b. Mark person j infected iff $\sum_i S_{j,i} > Z$ (threshold).

	🧪	🧪	🧪	🧪	🧪	...	🧪	$\sum_i S_{j,i}$
Amin	+0.3	+0.3	+0.3	−0.2	−0.2	...	+0.3	0
Boris	−1.0	+0.3	−1.0	+0.7	+0.7	...	−1.0	0
Caroline	+0.3	+0.3	+0.3	+0.7	−0.2	...	+0.3	0
David	+0.3	+0.3	−1.0	+0.7	+0.7	...	+0.3	0
Eve	+0.3	+0.3	+0.3	−0.2	−0.2	...	+0.3	0
Fred	−1.0	+0.3	−1.0	−0.2	−0.2	...	+0.3	0
Gábor	+0.3	+0.3	−1.0	−0.2	−0.2	...	+0.3	0
Henry	+0.3	+0.3	+0.3	−0.2	+0.7	...	+0.3	0
Results	0	0	0	1	1	...	0	

Infected $= \{$Caroline, Eve, Henry$\}$

Example: Scores

2b. Mark person j infected iff $\sum_i S_{j,i} > Z$ (threshold).

	🧪	🧪	🧪	🧪	🧪	...	🧪	$\sum_i S_{j,i}$
Amin	$+0.3$	$+0.3$	$+0.3$	-0.2	-0.2	...	$+0.3$	-5
Boris	-1.0	$+0.3$	-1.0	$+0.7$	$+0.7$	...	-1.0	-12
Caroline	$+0.3$	$+0.3$	$+0.3$	$+0.7$	-0.2	...	$+0.3$	$+41$
David	$+0.3$	$+0.3$	-1.0	$+0.7$	$+0.7$	...	$+0.3$	-3
Eve	$+0.3$	$+0.3$	$+0.3$	-0.2	-0.2	...	$+0.3$	$+38$
Fred	-1.0	$+0.3$	-1.0	-0.2	-0.2	...	$+0.3$	$+10$
Gábor	$+0.3$	$+0.3$	-1.0	-0.2	-0.2	...	$+0.3$	-1
Henry	$+0.3$	$+0.3$	$+0.3$	-0.2	$+0.7$	...	$+0.3$	$+40$
Results	0	0	0	1	1	...	0	

Infected $=$ {Caroline, Eve, Henry}

Example: Scores

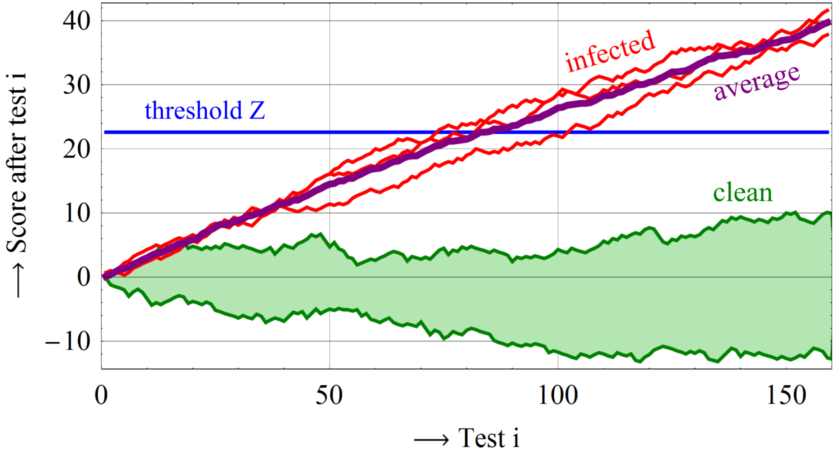
2b. Mark person j infected iff $\sum_i S_{j,i} > Z$ (threshold).

	🧪	🧪	🧪	🧪	🧪	...	🧪	$\sum_i S_{j,i}$
Amin	+0.3	+0.3	+0.3	−0.2	−0.2	...	+0.3	−5
Boris	−1.0	+0.3	−1.0	+0.7	+0.7	...	−1.0	−12
Caroline	+0.3	+0.3	+0.3	+0.7	−0.2	...	+0.3	+41
David	+0.3	+0.3	−1.0	+0.7	+0.7	...	+0.3	−3
Eve	+0.3	+0.3	+0.3	−0.2	−0.2	...	+0.3	+38
Fred	−1.0	+0.3	−1.0	−0.2	−0.2	...	+0.3	+10
Gábor	+0.3	+0.3	−1.0	−0.2	−0.2	...	+0.3	−1
Henry	+0.3	+0.3	+0.3	−0.2	+0.7	...	+0.3	+40
Results	0	0	0	1	1	...	0	

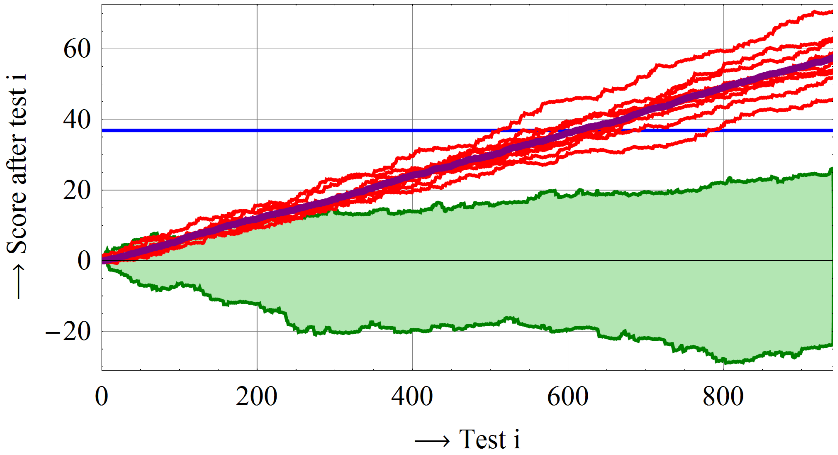
$$\text{Infected} = \{\text{Caroline}, \text{Eve}, \text{Henry}\}$$

$$\text{Marked} = \{\text{Caroline}, \text{Eve}, \text{Henry}\}$$

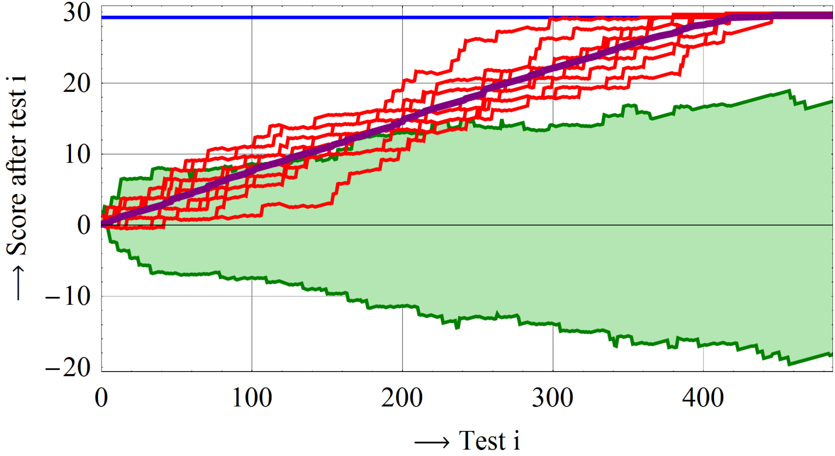
Example: Scores



Larger Example: Non-Adaptive



Larger Example: Adaptive



Framework: Other Models

Traditional group testing

- Positive test result iff at least one infected is tested

Framework: Other Models

Traditional group testing

- Positive test result iff at least one infected is tested

Noisy group testing

- Dilution: Clean sample might test positive
- Additive: Infected sample might test negative
- Combined: Any sample might have a wrong test result
- ...

Framework: Other Models

Traditional group testing

- Positive test result iff at least one infected is tested

Noisy group testing

- Dilution: Clean sample might test positive
- Additive: Infected sample might test negative
- Combined: Any sample might have a wrong test result
- ...

Threshold group testing

- Majority: Positive result iff more than ℓ infected
- Bernoulli: Few infected tested leads to random result
- Linear: More infected tested leads to more positive results
- ...

Framework: Other Models

Traditional group testing

- Tests required: $T \sim 2K \ln N$

Noisy group testing

- Dilution: $T \sim 2K \ln N/(1-r)$
- Additive: $T \sim 2K \ln N/(1-\sqrt{2r})$
- Combined: $T \sim 2K \ln N/(1-\sqrt{2r})$
- ...

Threshold group testing

- Majority: $T \sim \pi K \ln N$
- Bernoulli: $T \sim 4K \ln N$
- Linear: $T \sim 2K^2 \ln N$
- ...

Framework: Versatility

Performance

- Small number of tests (asymptotically)
- Low complexity decoding step (simple decoder)

Framework: Versatility

Performance

- Small number of tests (asymptotically)
- Low complexity decoding step (simple decoder)

Different models

- Can be applied to arbitrary models
- Wrong model does not increase false positives

Framework: Versatility

Performance

- Small number of tests (asymptotically)
- Low complexity decoding step (simple decoder)

Different models

- Can be applied to arbitrary models
- Wrong model does not increase false positives

Adaptive construction

- Number of tests significantly decreases adaptively
- Adaptive: X can still be generated in advance*
- Adaptive: Can deal with unknown K efficiently
- Semi-adaptive: $O(\log N)$ rounds, $O(K)$ tests/round

Conclusion

Framework for probabilistic group testing

- Score-based construction
- Explicit procedure for obtaining optimal parameters
- Versatile construction

Results when applied to common models:

- Traditional model: $T \sim 2K \ln N$
- Dilution noise: $T \sim 2K \ln N/(1-r)$
- Additive noise: $T \sim 2K \ln N/(1-\sqrt{2r})$
- Threshold models: $T = \Theta(K \ln N)$
- Can be applied to arbitrary models

Questions?