HYPER: Flexible and effective pooled testing via hypergraph factorization


David Hong
U Penn


Rounak Dey
Harvard U


Xihong Lin
Harvard U


Brian Cleary
Broad Institute


Edgar Dobriban
U Penn

Work supported in part by the Dean's Fund for Postdoctoral Research of the Wharton School, and a grant from the Partners in Health.

Pooled testing

Goal: Screen more people using fewer tests (given limited/constrained resources).

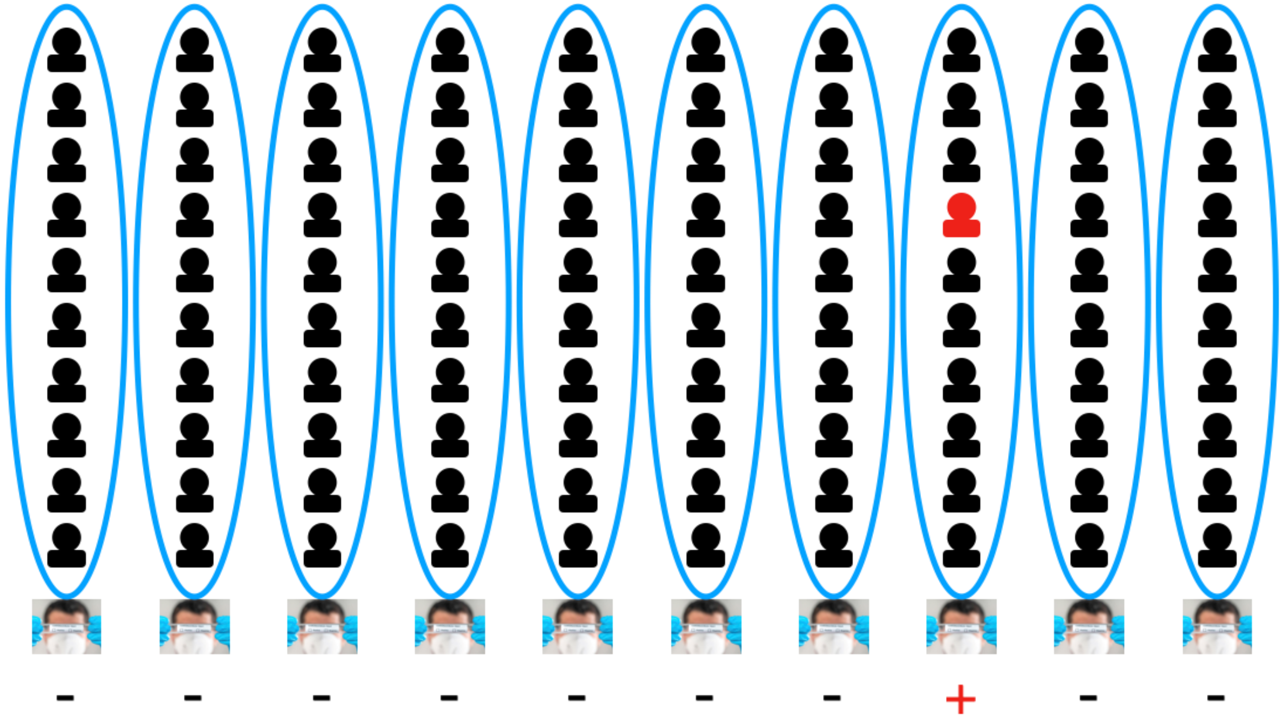
Images courtesy: https://www.math.uwaterloo.ca/~rwoldfor/COVID19/poolingSamples.html

Pooled testing via Dorfman pooling

Goal: Screen more people using fewer tests (given limited/constrained resources).

Classical Dorfman Pooling (1943)

Stage 1: pooled testing ($n = 100$ individuals in $m = 10$ pools)



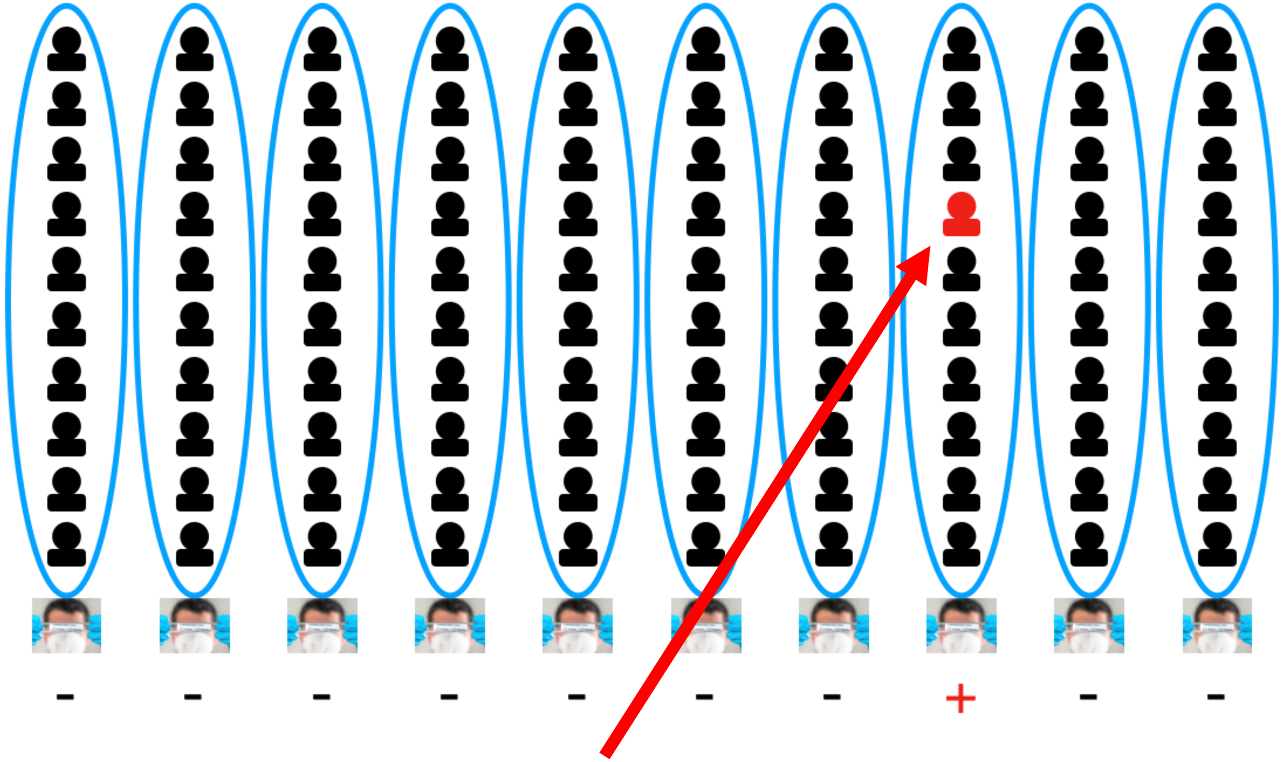
Images courtesy: https://www.math.uwaterloo.ca/~rwoldfor/COVID19/poolingSamples.html

Pooled testing via Dorfman pooling

Goal: Screen more people using fewer tests (given limited/constrained resources).

Classical Dorfman Pooling

Stage 1: pooled testing ($n = 100$ individuals in $m = 10$ pools)



1 infected individual
(prevalence $p = 1\%$)

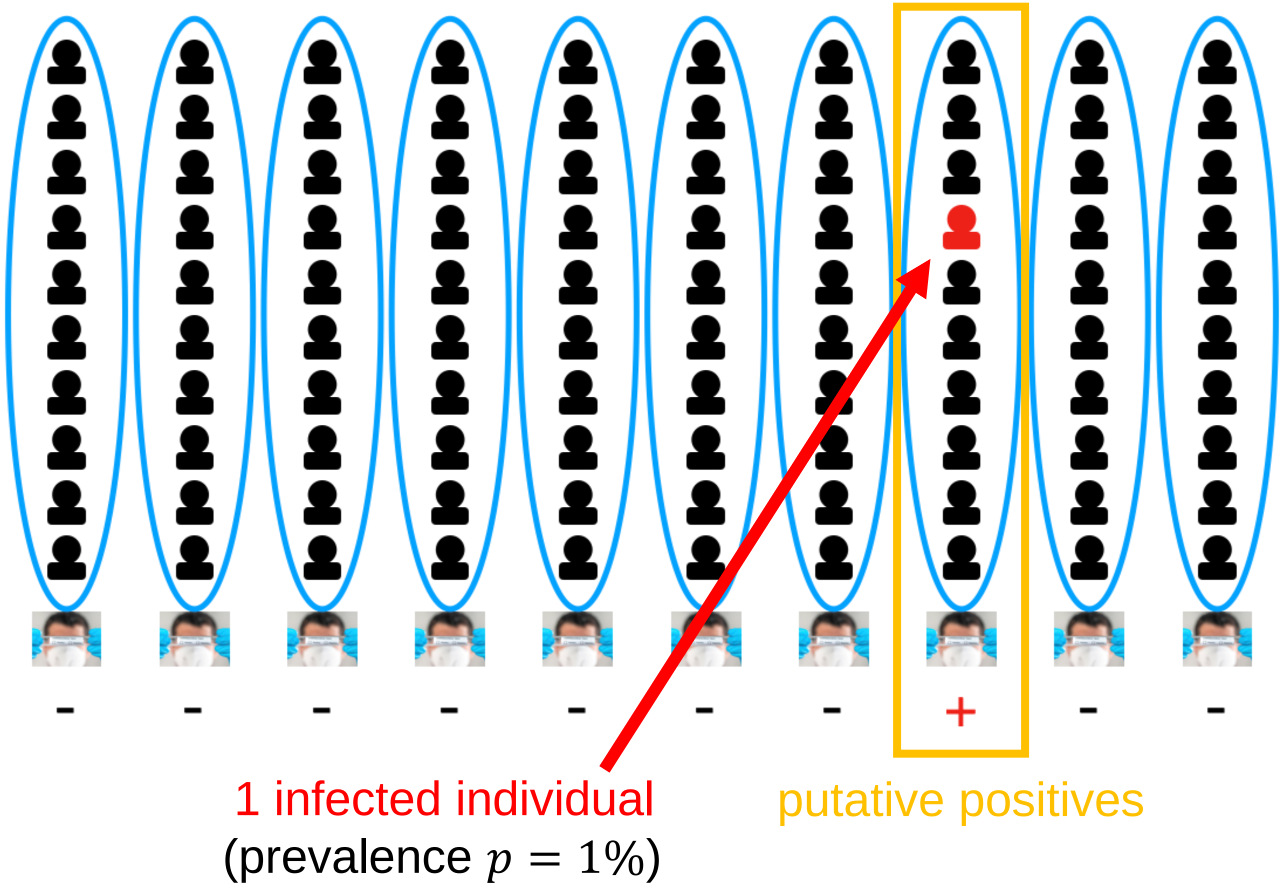
Images courtesy: https://www.math.uwaterloo.ca/~rwoldfor/COVID19/poolingSamples.html

Pooled testing via Dorfman pooling

Goal: Screen more people using fewer tests (given limited/constrained resources).

Classical Dorfman Pooling

Stage 1: pooled testing ($n = 100$ individuals in $m = 10$ pools)



Images courtesy: https://www.math.uwaterloo.ca/~rwoldfor/COVID19/poolingSamples.html

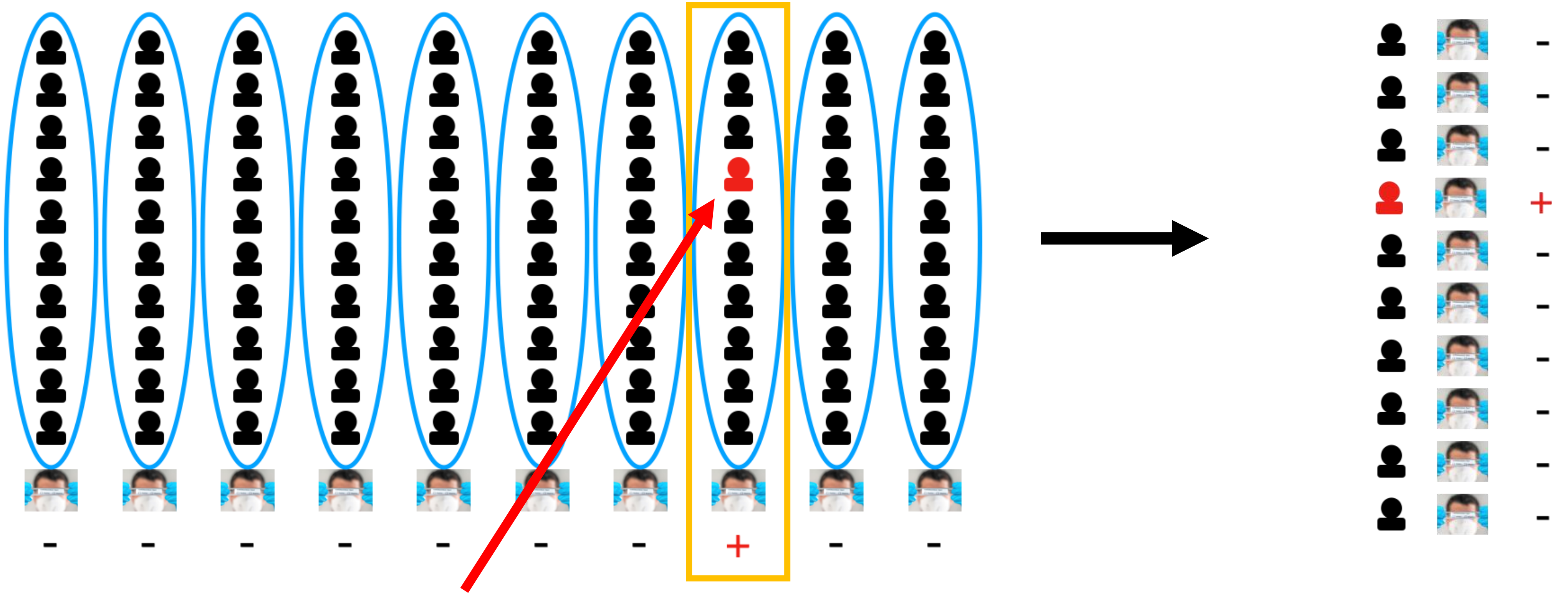
Pooled testing via Dorfman pooling

Goal: Screen more people using fewer tests (given limited/constrained resources).

Classical Dorfman Pooling

Stage 1: pooled testing ($n = 100$ individuals in $m = 10$ pools)

Stage 2: test putative positives



1 infected individual
(prevalence $p = 1\%$)

putative positives

Images courtesy: https://www.math.uwaterloo.ca/~rwoldfor/COVID19/poolingSamples.html

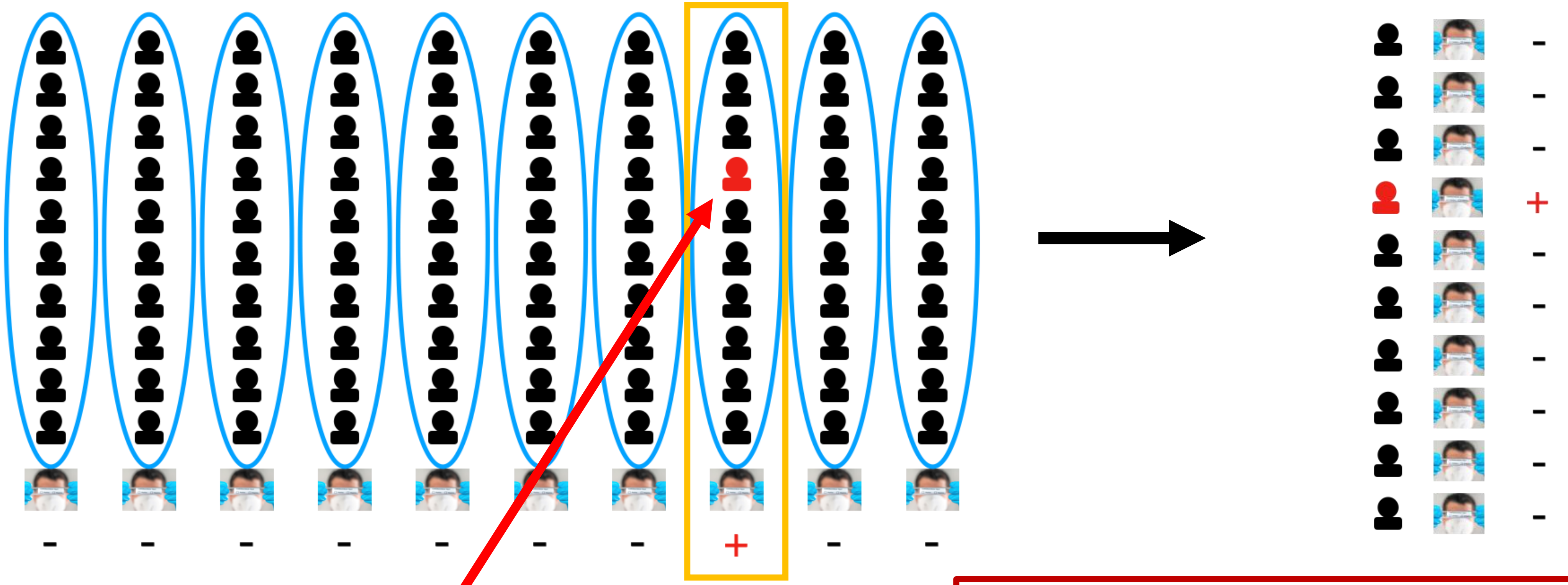
Pooled testing via Dorfman pooling

Goal: Screen more people using fewer tests (given limited/constrained resources).

Classical Dorfman Pooling

Stage 1: pooled testing ($n = 100$ individuals in $m = 10$ pools)

Stage 2: test putative positives



1 infected individual
(prevalence $p = 1\%$)

putative positives

indiv. screened: 100
tests used: 10 (stage 1) + 10 (stage 2) = 20

Images courtesy: https://www.math.uwaterloo.ca/~rwoldfor/COVID19/poolingSamples.html

Dorfman pooling in action at the Broad!

NEWS / 02.25.21

EMAIL

Broad Institute is processing pooled COVID-19 tests for Massachusetts K-12 schools

By Calley Jones

Here's more on how pooled testing works, and what students and parents can expect.

Credit : Scott Sassone, Broad Communications
Meghan Gillespie processes a pooled sample at Broad's COVID-19 diagnostic facility.

Pooled testing involves placing nasal swabs from multiple people into the same sample tube and processing each tube as a single sample, saving time and resources. Credit: Scott Sassone, Broad Communications

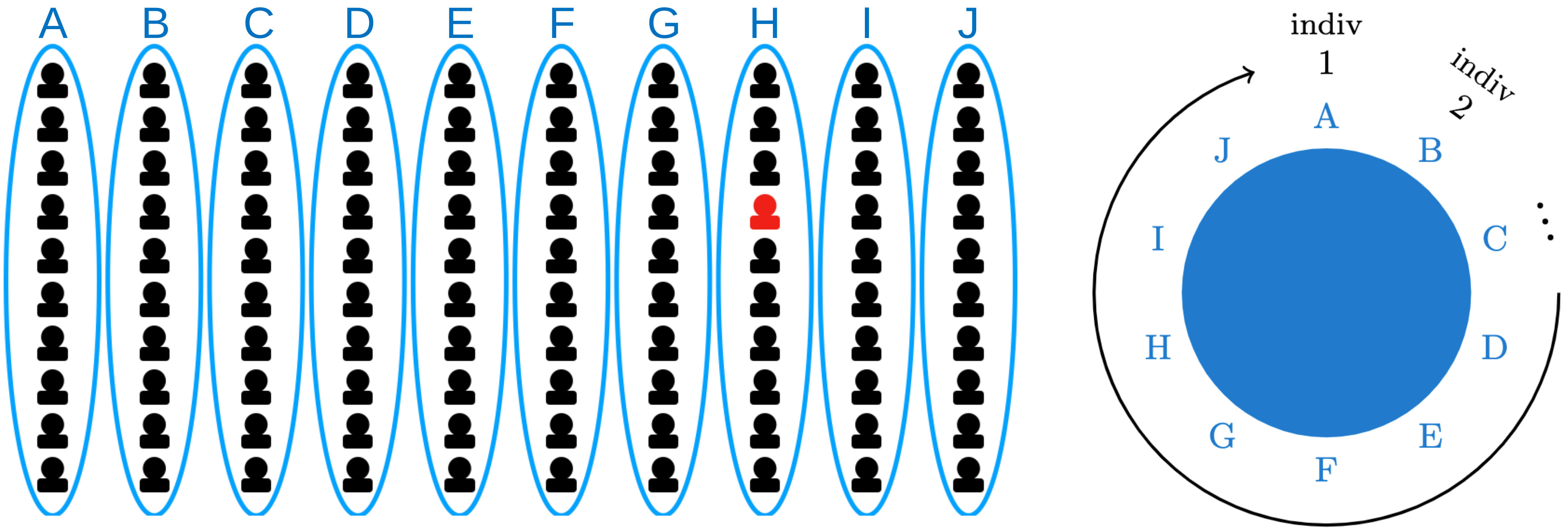
"So far, Broad has processed 15,000 pools […] representing approximately 84,000 people […]."

"Five to ten individual swabs […] are collected in the same tube […]. If a pool is found to be positive for the virus, the school re-tests people individually in that pool to identify who has the virus."

Source: https://www.broadinstitute.org/news/broad-institute-processing-pooled-covid-19-tests-massachusetts-k-12-schools

Closer look: how does Dorfman assign individuals to pools?

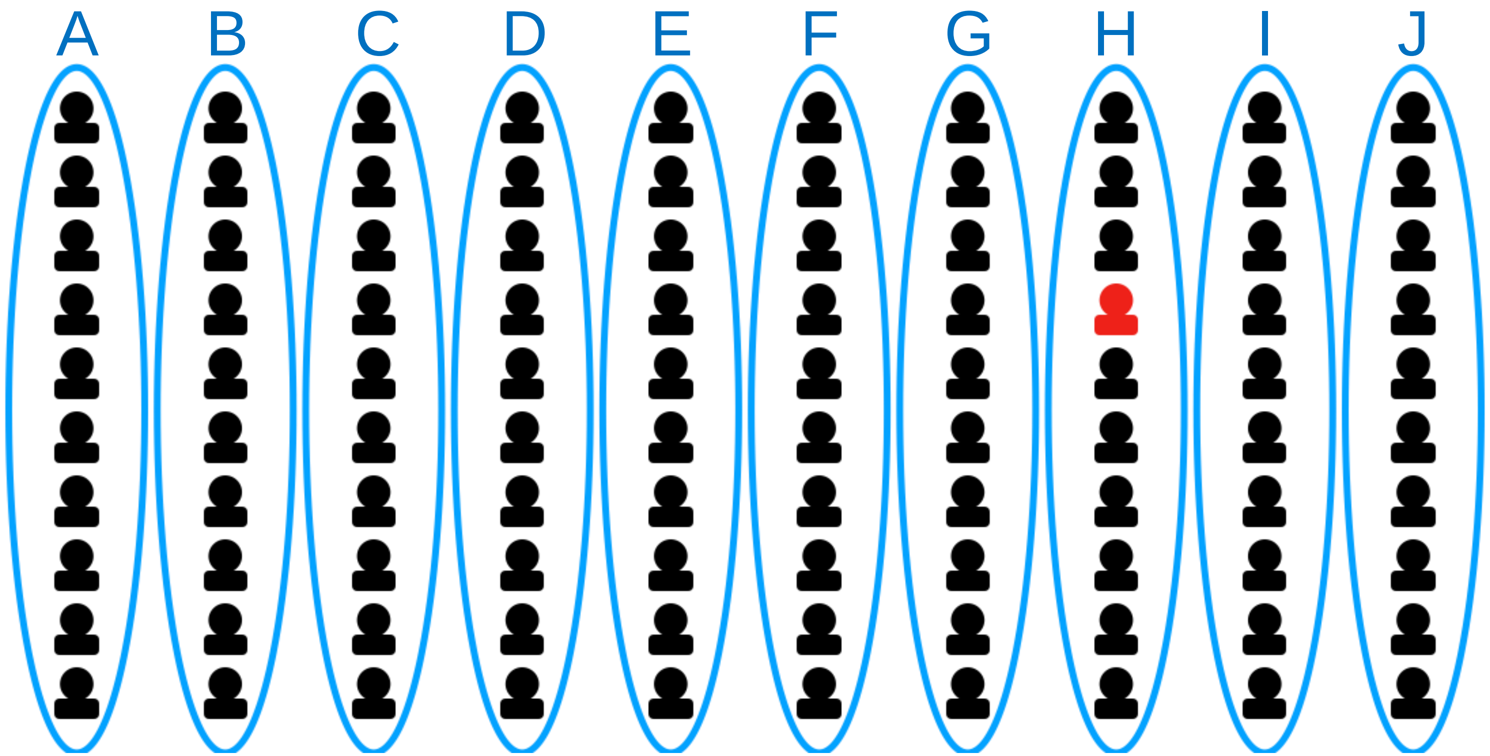
Each individual is assigned to $q = 1$ of the pools by cycling through the list of all pools.

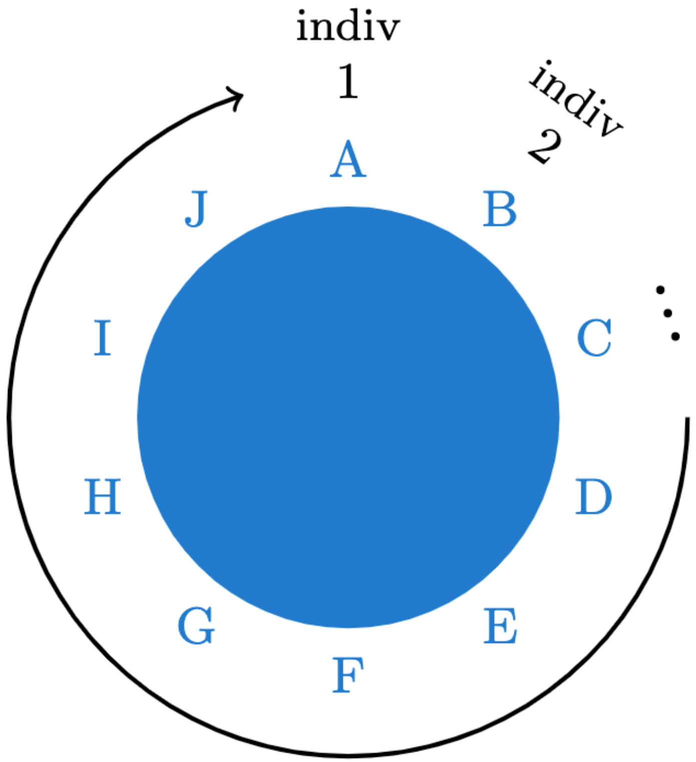


Individual	1	2	3	4	5	6	7	8	9	10	11	12	…
Pool	A	B	C	D	E	F	G	H	I	J	A	B	…

Closer look: how does Dorfman assign individuals to pools?

Each individual is assigned to $q = 1$ of the pools by cycling through the list of all pools.



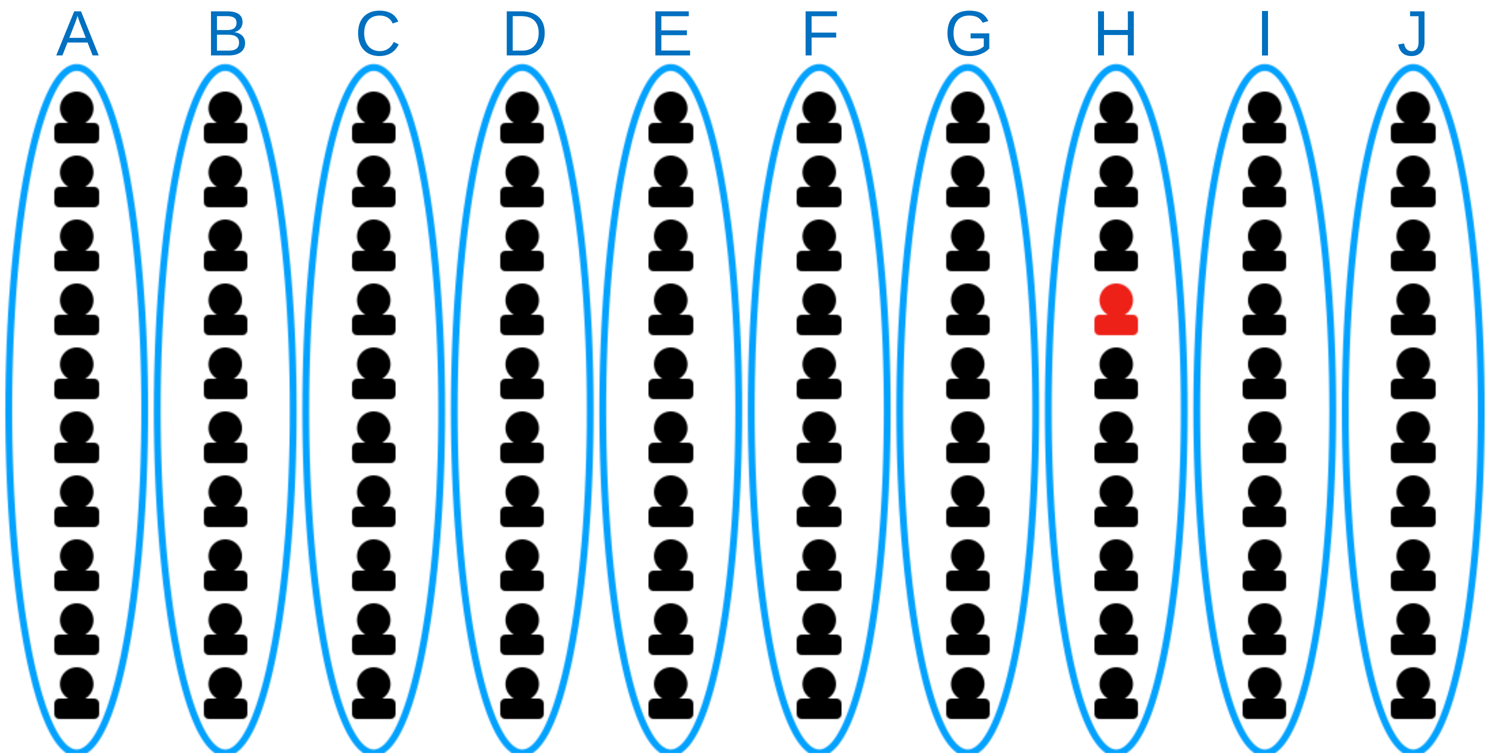


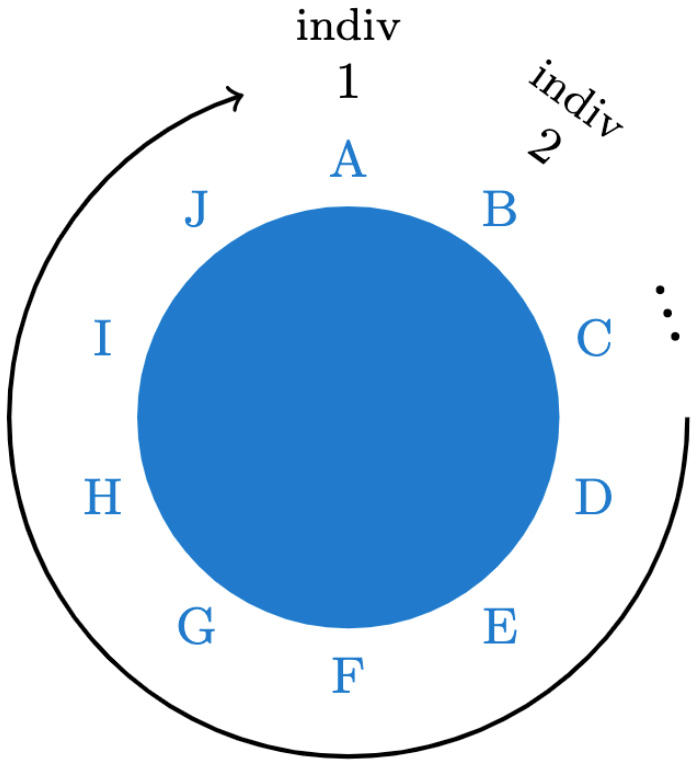
Individual	1	2	3	4	5	6	7	8	9	10	11	12	…
Pool	A	B	C	D	E	F	G	H	I	J	A	B	…

Advantage: Simple and balanced – facilitates robust implementation

Closer look: how does Dorfman assign individuals to pools?

Each individual is assigned to $q = 1$ of the pools by cycling through the list of all pools.





Individual	1	2	3	4	5	6	7	8	9	10	11	12	…
Pool	A	B	C	D	E	F	G	H	I	J	A	B	…

Advantage: Simple and balanced – facilitates robust implementation

Limitation: Suboptimal – efficiency can be improved using designs with $q > 1$

More efficient pooling: extending the Dorfman design?

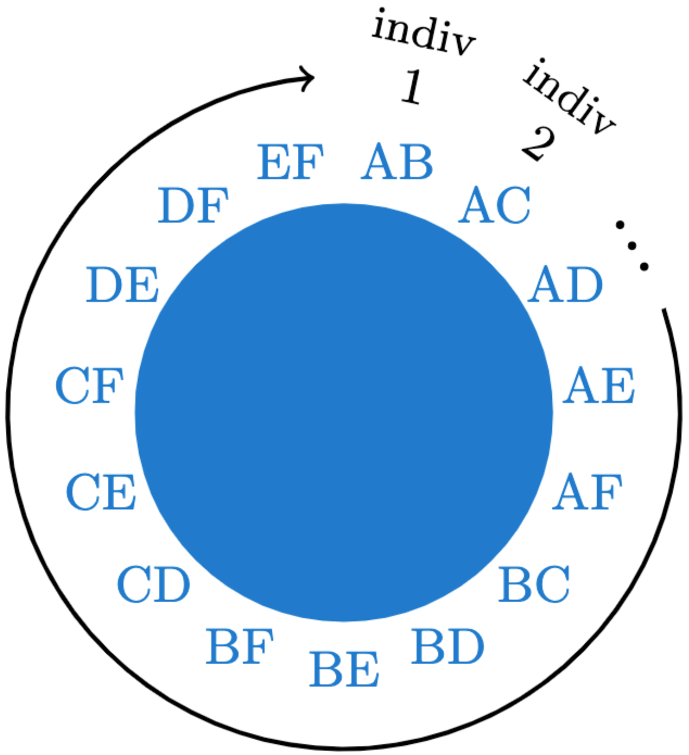
Question: How to allow individuals to be assigned to multiple pools ($q > 1$)?

More efficient pooling: extending the Dorfman design?

Question: How to allow individuals to be assigned to multiple pools ($q > 1$)?

Naïve approach for assigning each person to two pools ($q = 2$):

1. Order all possible pool pairs (AB,AC,BC,DE,…) lexicographically.
2. Cycle through the ordered pool pairs to assign the individuals.



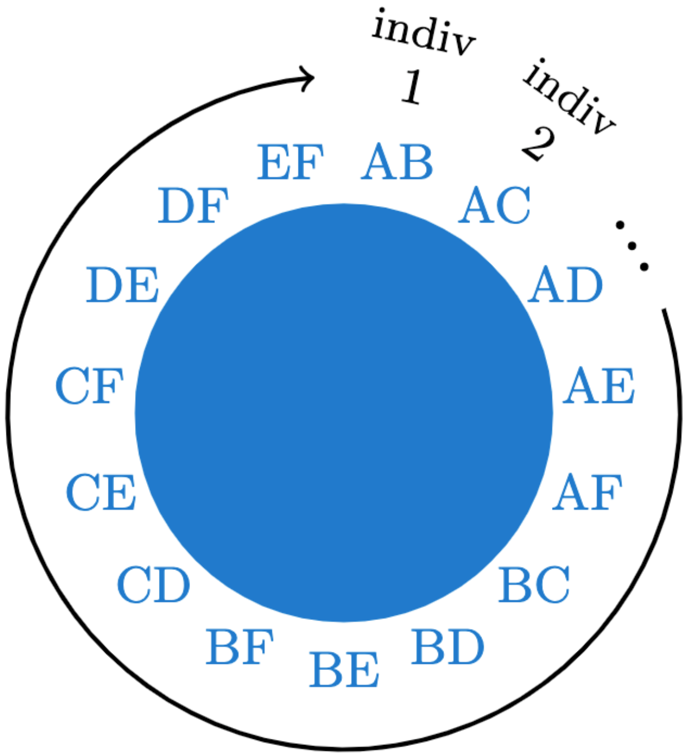
Individual	1	2	3	4	5	6	7	8
Pool	AB	AC	AD	AE	AF	BC	BD	BE

More efficient pooling: extending the Dorfman design?

Question: How to allow individuals to be assigned to multiple pools ($q > 1$)?

Naïve approach for assigning each person to two pools ($q = 2$):

1. Order all possible pool pairs (AB,AC,BC,DE,…) lexicographically.
2. Cycle through the ordered pool pairs to assign the individuals.



Individual	1	2	3	4	5	6	7	8
Pool	AB	AC	AD	AE	AF	BC	BD	BE

Problem: Pools are not balanced!

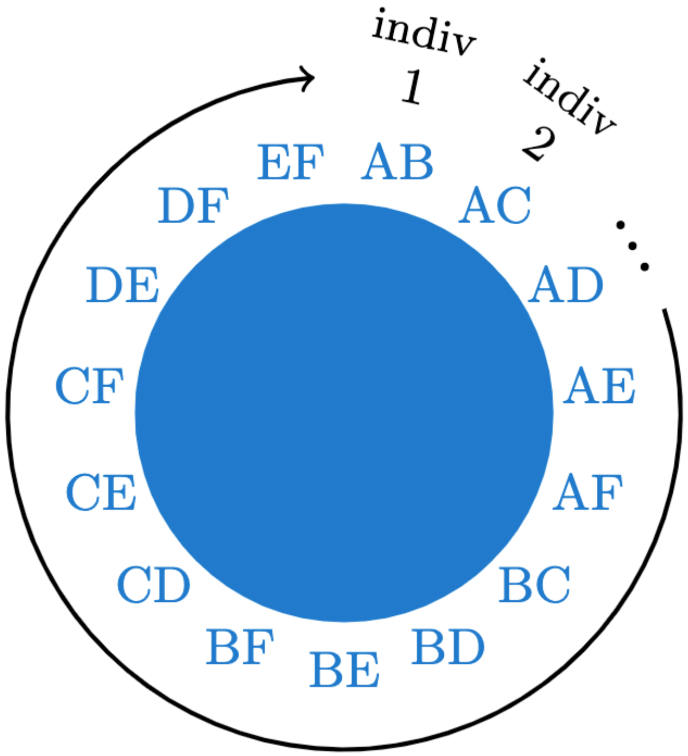
A=(1,2,3,4,5) B=(1,6,7,8) C=(2,6)

D=(3,7) E=(4,8) F=(5)

More efficient pooling: extending the Dorfman design?

Question: How to allow individuals to be assigned to multiple pools ($q > 1$)?

Naïve approach for assigning each person to two pools ($q = 2$):
1. Order all possible pool pairs (AB,AC,BC,DE,…) lexicographically.
2. Cycle through the ordered pool pairs to assign the individuals.



Individual	1	2	3	4	5	6	7	8
Pool	AB	AC	AD	AE	AF	BC	BD	BE

Problem: Pools are not balanced!

A=(1,2,3,4,5) B=(1,6,7,8) C=(2,6)
D=(3,7) E=(4,8) F=(5)

→ uneven dilution
→ uneven sensitivity (indiv 1 has higher risk of false neg)

More efficient pooling: extending the Dorfman design?

Tremendous study and progress on pooled testing in general – more than we can review today!

More efficient pooling: extending the Dorfman design?

Tremendous study and progress on pooled testing in general – more than we can review today!

Today's goal. Assign individuals to pools in a maximally balanced way:

More efficient pooling: extending the Dorfman design?

Tremendous study and progress on pooled testing in general – more than we can review today!

Today's goal. Assign individuals to pools in a maximally balanced way:

1. Assign all individuals to the same number q of pools

Individual	1	2	3	4	5	6
Pool	AB	C	DE	F	AB	CD

✗

Individual	1	2	3	4	5	6
Pool	AB	CD	EF	BC	DF	AE

✓

More efficient pooling: extending the Dorfman design?

Tremendous study and progress on pooled testing in general – more than we can review today!

Today's goal. Assign individuals to pools in a maximally balanced way:

1. Assign all individuals to the same number q of pools

Individual	1	2	3	4	5	6
Pool	AB	C	DE	F	AB	CD

✗

Individual	1	2	3	4	5	6
Pool	AB	CD	EF	BC	DF	AE

✓

2. Assign the m pools as evenly as possible

Individual	1	2	3	4	5	6
Pool	AB	AC	AD	AE	AF	BC

✗

Individual	1	2	3	4	5	6
Pool	AB	CD	EF	BC	DF	AE

✓

More efficient pooling: extending the Dorfman design?

Tremendous study and progress on pooled testing in general – more than we can review today!

Today's goal. Assign individuals to pools in a maximally balanced way:

1. Assign all individuals to the same number q of pools

Individual	1	2	3	4	5	6
Pool	AB	C	DE	F	AB	CD

✗

Individual	1	2	3	4	5	6
Pool	AB	CD	EF	BC	DF	AE

✓

2. Assign the m pools as evenly as possible

Individual	1	2	3	4	5	6
Pool	AB	AC	AD	AE	AF	BC

✗

Individual	1	2	3	4	5	6
Pool	AB	CD	EF	BC	DF	AE

✓

3. Assign the $\binom{m}{q}$ possible pool combinations as evenly as possible

Individual	1	2	3	4	5	6
Pool	AB	CD	EF	AB	CD	EF

✗

Individual	1	2	3	4	5	6
Pool	AB	CD	EF	BC	DF	AE

✓

More efficient pooling: extending the Dorfman design?

Tremendous study and progress on pooled testing in general – more than we can review today!

Today's goal. Assign individuals to pools in a maximally balanced way:

1. Assign all individuals to the same number q of pools

Individual	1	2	3	4	5	6
Pool	AB	C	DE	F	AB	CD

✗

Individual	1	2	3	4	5	6
Pool	AB	CD	EF	BC	DF	AE

✓

2. Assign the m pools as evenly as possible

Individual	1	2	3	4	5	6
Pool	AB	AC	AD	AE	AF	BC

✗

Individual	1	2	3	4	5	6
Pool	AB	CD	EF	BC	DF	AE

✓

3. Assign the $\binom{m}{q}$ possible pool combinations as evenly as possible

Individual	1	2	3	4	5	6
Pool	AB	CD	EF	AB	CD	EF

✗

Individual	1	2	3	4	5	6
Pool	AB	CD	EF	BC	DF	AE

✓

…in a way that is easy to adapt/tailor (e.g., easily increase # indiv in batch).

Approach 1: random designs

Random assignment: assign each individual to q pools uniformly at random.

Approach 1: random designs

Random assignment: assign each individual to q pools uniformly at random.

Advantage: pools and pool combinations are balanced on average

$$\Pr(\mathrm{A\_}) = \Pr(\mathrm{B\_}) = \Pr(\mathrm{C\_}) = \Pr(\mathrm{D\_}) = \Pr(\mathrm{E\_}) = \Pr(\mathrm{F\_}) = 5/15$$
$$\Pr(\mathrm{AB}) = \Pr(\mathrm{AC}) = \Pr(\mathrm{AD}) = \cdots = \Pr(\mathrm{DF}) = \Pr(\mathrm{EF}) = 1/15$$

Approach 1: random designs

Random assignment: assign each individual to q pools uniformly at random.

Advantage: pools and pool combinations are balanced on average

$$\Pr(\mathrm{A\_}) = \Pr(\mathrm{B\_}) = \Pr(\mathrm{C\_}) = \Pr(\mathrm{D\_}) = \Pr(\mathrm{E\_}) = \Pr(\mathrm{F\_}) = 5/15$$
$$\Pr(\mathrm{AB}) = \Pr(\mathrm{AC}) = \Pr(\mathrm{AD}) = \cdots = \Pr(\mathrm{DF}) = \Pr(\mathrm{EF}) = 1/15$$

Limitation: random draws are themselves rarely maximally balanced

Approach 1: random designs

Random assignment: assign each individual to q pools uniformly at random.

Advantage: pools and pool combinations are balanced on average

$$\Pr(\mathrm{A\_}) = \Pr(\mathrm{B\_}) = \Pr(\mathrm{C\_}) = \Pr(\mathrm{D\_}) = \Pr(\mathrm{E\_}) = \Pr(\mathrm{F\_}) = 5/15$$
$$\Pr(\mathrm{AB}) = \Pr(\mathrm{AC}) = \Pr(\mathrm{AD}) = \cdots = \Pr(\mathrm{DF}) = \Pr(\mathrm{EF}) = 1/15$$

Limitation: random draws are themselves rarely maximally balanced

Individual	1	2	3	4	5	6	7	8
Pool	CD	EF	BE	CD	DE	CF	BE	BC

Approach 1: random designs

Random assignment: assign each individual to q pools uniformly at random.

Advantage: pools and pool combinations are balanced on average

$$\Pr(A\_) = \Pr(B\_) = \Pr(C\_) = \Pr(D\_) = \Pr(E\_) = \Pr(F\_) = 5/15$$
$$\Pr(AB) = \Pr(AC) = \Pr(AD) = \cdots = \Pr(DF) = \Pr(EF) = 1/15$$

Limitation: random draws are themselves rarely maximally balanced

Individual	1	2	3	4	5	6	7	8
Pool	CD	EF	BE	CD	DE	CF	BE	BC

A=(); B=(3,7,8); C=(1,4,6,8); D=(1,4,5); E=(2,3,5,7); F=(2,6)

AB=(); AC=(); AD=(); AE=(); AF=(); BC=(8); BD=(); BE=(3,7);
BF=(); CD=(1,4); CE=(); CF=(6); DE=(5); DF=(); EF=(2)

Approach 1: random designs

Random assignment: assign each individual to q pools uniformly at random.

Advantage: pools and pool combinations are balanced on average

$$\Pr(A\_) = \Pr(B\_) = \Pr(C\_) = \Pr(D\_) = \Pr(E\_) = \Pr(F\_) = 5/15$$
$$\Pr(AB) = \Pr(AC) = \Pr(AD) = \cdots = \Pr(DF) = \Pr(EF) = 1/15$$

Limitation: random draws are themselves rarely maximally balanced

Individual	1	2	3	4	5	6	7	8
Pool	CD	EF	BE	CD	DE	CF	BE	BC

A=(); B=(3,7,8); C=(1,4,6,8); D=(1,4,5); E=(2,3,5,7); F=(2,6)

AB=(); AC=(); AD=(); AE=(); AF=(); BC=(8); BD=(); BE=(3,7);
BF=(); CD=(1,4); CE=(); CF=(6); DE=(5); DF=(); EF=(2)

Only 1.3% are maximally balanced for this case (found by exhaustive search).

Approach 2: exhaustive search

Exhaustive search: search through all possible pool assignments

Approach 2: exhaustive search

Exhaustive search: search through all possible pool assignments

Advantage: systematic – guaranteed to find a maximally balanced design

Approach 2: exhaustive search

Exhaustive search: search through all possible pool assignments

Advantage: systematic – guaranteed to find a maximally balanced design

Limitation: only practical for small cases.

How big is the search space?

$$\#\text{ possible assignments} = \binom{m}{q}^{n}$$

Approach 2: exhaustive search

Exhaustive search: search through all possible pool assignments

Advantage: systematic – guaranteed to find a maximally balanced design

Limitation: only practical for small cases.

How big is the search space?

$$\#\ \text{possible assignments} = \binom{m}{q}^{n}$$

Can be reduced by skipping designs that are “obviously unbalanced”…but still large.

Approach 2: exhaustive search

Exhaustive search: search through all possible pool assignments

Advantage: systematic – guaranteed to find a maximally balanced design

Limitation: only practical for small cases.

How big is the search space?

$$\#\ \text{possible assignments} = \binom{m}{q}^n$$

Can be reduced by skipping designs that are “obviously unbalanced”…but still large.

Will also need to be repeated for any change in n, m or q → harder to adapt

Approach 2: exhaustive search

Exhaustive search: search through all possible pool assignments

Advantage: systematic – guaranteed to find a maximally balanced design

Limitation: only practical for small cases.

How big is the search space?

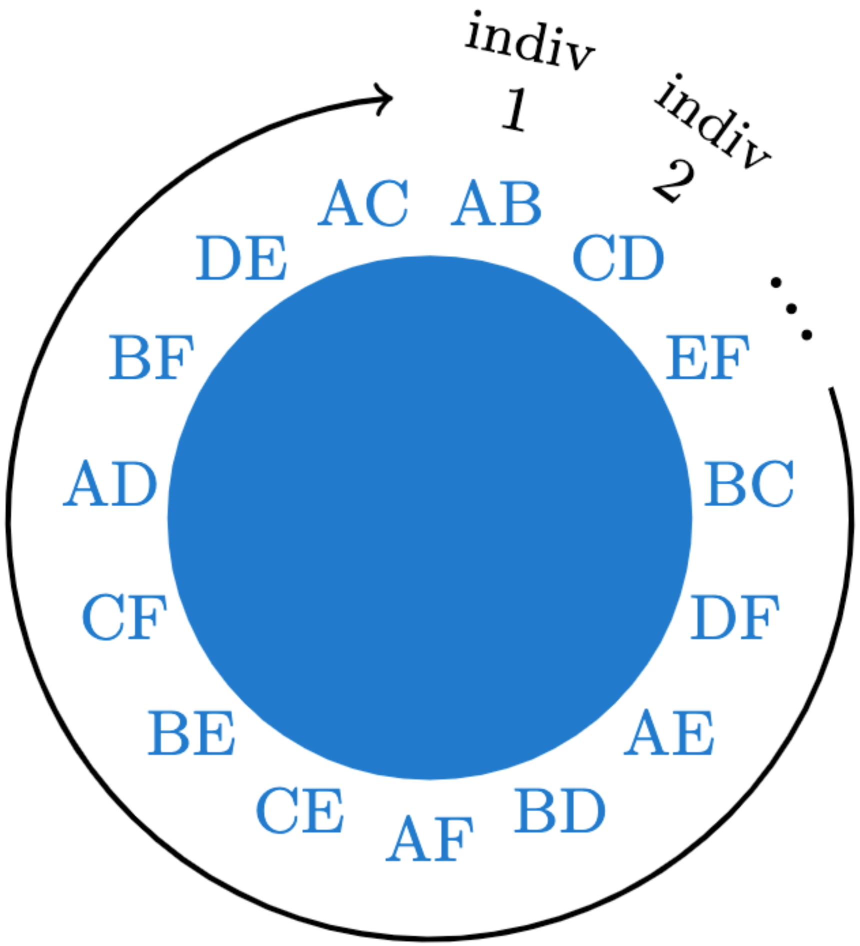
$$\#\ \text{possible assignments} = \binom{m}{q}^n$$

Can be reduced by skipping designs that are "obviously unbalanced"…but still large.

Will also need to be repeated for any change in n, m or q → harder to adapt

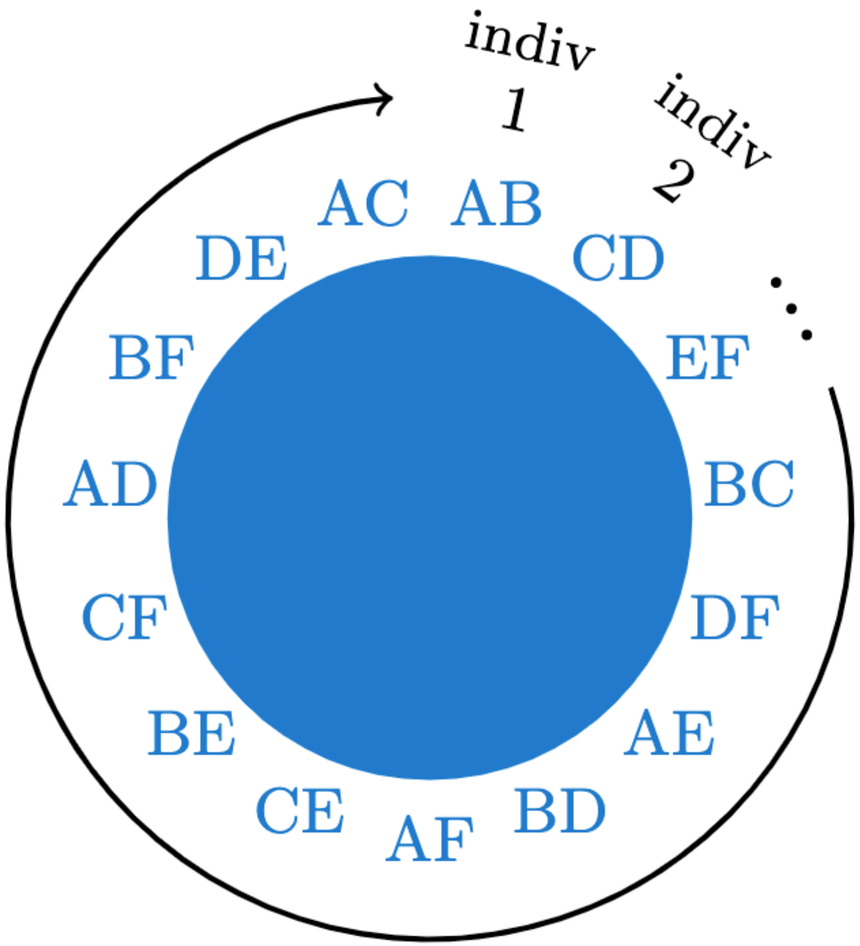
Do maximally balanced designs even exist in general?

Design given by HYPER



Individual	1	2	3	4	5	6
Pool	AB	CD	EF	BC	DF	AE

Design given by HYPER

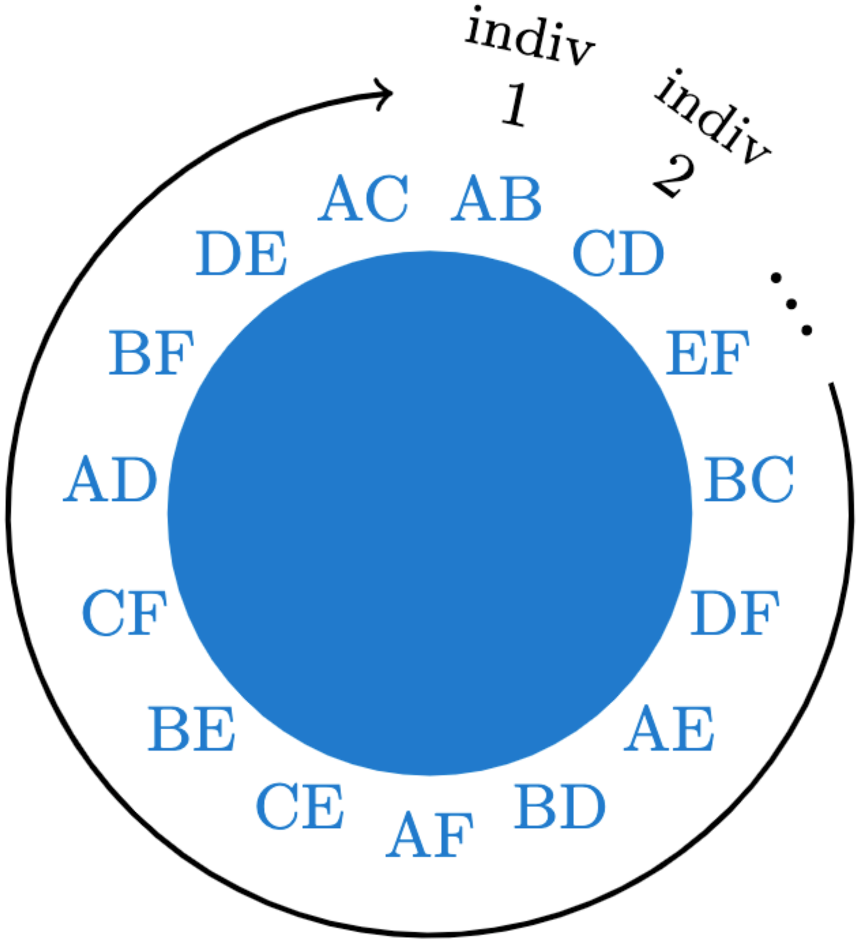


Individual	1	2	3	4	5	6
Pool	AB	CD	EF	BC	DF	AE

A=(1,6); B=(1,4); C=(2,4); D=(2,5); E=(3,6); F=(3,5)

AB=(1); AC=(); AD=(); AE=(6); AF=(); BC=(4); BD=(); BE=(); BF=(); CD=(2); CE=(); CF=(); DE=(); DF=(5); EF=(3)

Design given by HYPER



Individual	1	2	3	4	5	6
Pool	AB	CD	EF	BC	DF	AE

A=(1,6); B=(1,4); C=(2,4); D=(2,5); E=(3,6); F=(3,5)

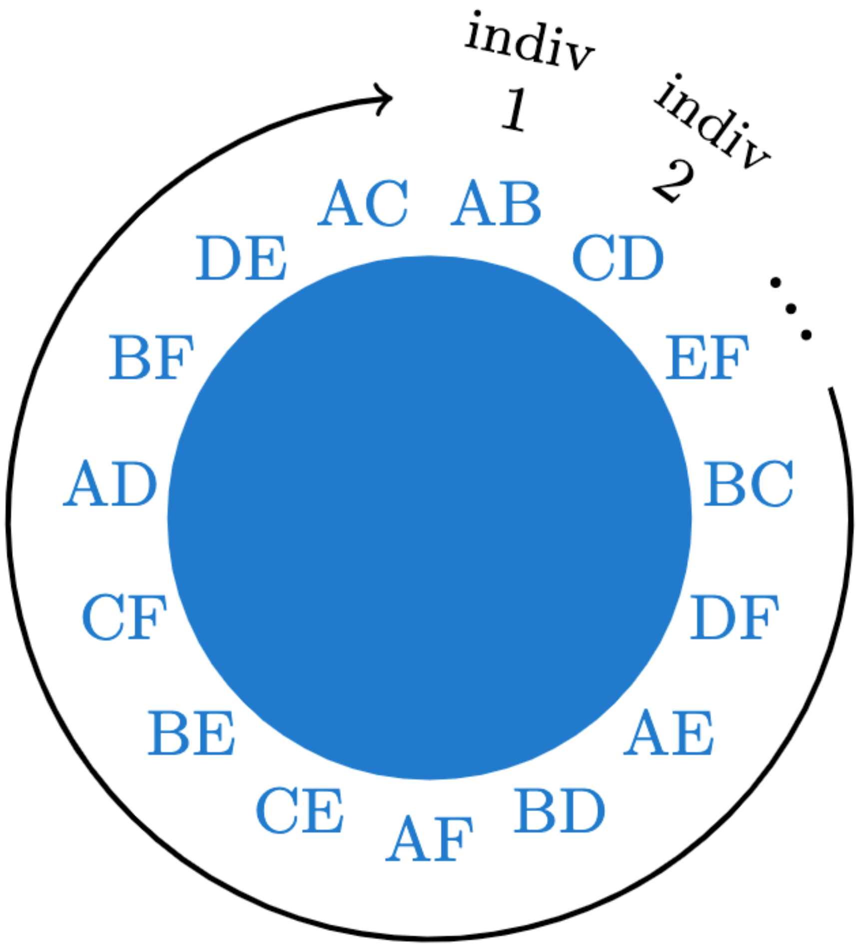
AB=(1); AC=(); AD=(); AE=(6); AF=(); BC=(4); BD=(); BE=(); BF=(); CD=(2); CE=(); CF=(); DE=(); DF=(5); EF=(3)

Individual	1	2	3	4	5	6	7	8
Pool	AB	CD	EF	BC	DF	AE	BD	AF

A=(1,6,8); B=(1,4,7); C=(2,4); D=(2,5,7); E=(3,6); F=(3,5,8)

AB=(1); AC=(); AD=(); AE=(6); AF=(8); BC=(4); BD=(7); BE=(); BF=(); CD=(2); CE=(); CF=(); DE=(); DF=(5); EF=(3)

Design given by HYPER



Individual	1	2	3	4	5	6
Pool	AB	CD	EF	BC	DF	AE

A=(1,6); B=(1,4); C=(2,4); D=(2,5); E=(3,6); F=(3,5)

AB=(1); AC=(); AD=(); AE=(6); AF=(); BC=(4); BD=(); BE=();
BF=(); CD=(2); CE=(); CF=(); DE=(); DF=(5); EF=(3)

Individual	1	2	3	4	5	6	7	8
Pool	AB	CD	EF	BC	DF	AE	BD	AF

A=(1,6,8); B=(1,4,7); C=(2,4); D=(2,5,7); E=(3,6); F=(3,5,8)

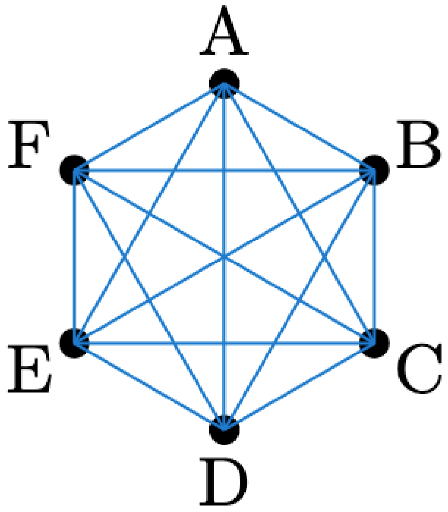
AB=(1); AC=(); AD=(); AE=(6); AF=(8); BC=(4); BD=(7); BE=();
BF=(); CD=(2); CE=(); CF=(); DE=(); DF=(5); EF=(3)

How did we get this design?

Another angle on the problem: hypergraph factorization

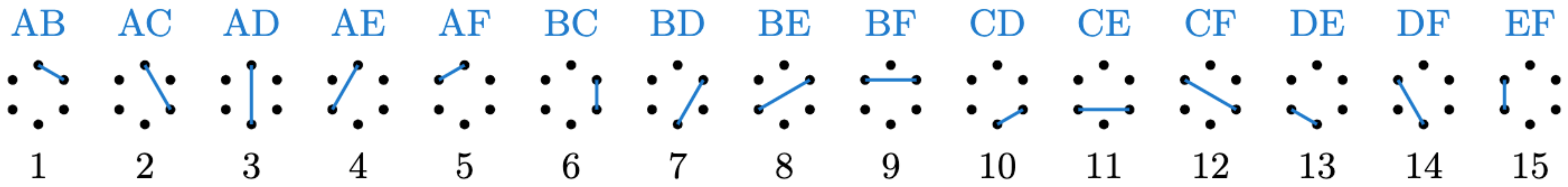
Idea: think of pools as vertices in a graph → pool assignments are (hyper)edges

K_6^2



vertices : pools (A-F)
edges : all possible indiv. pool assignments (AB,AC,...)

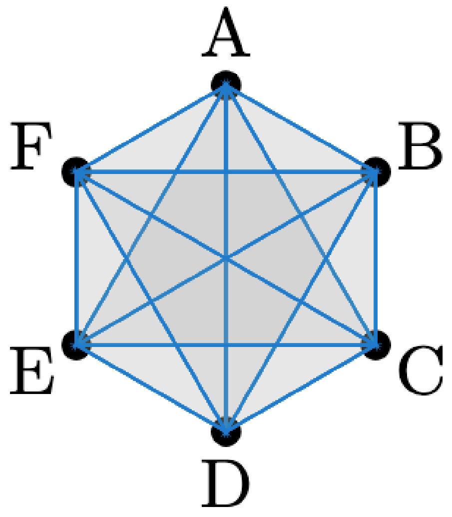
The 15 hyperedges of K_6^2 (in lexicographic order)



Another angle on the problem: hypergraph factorization

Idea: think of pools as vertices in a graph → pool assignments are hyperedges

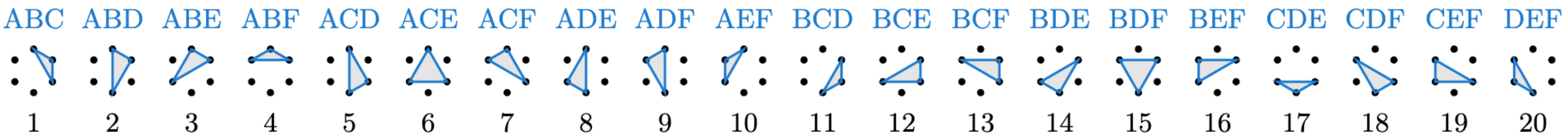
K_6^3



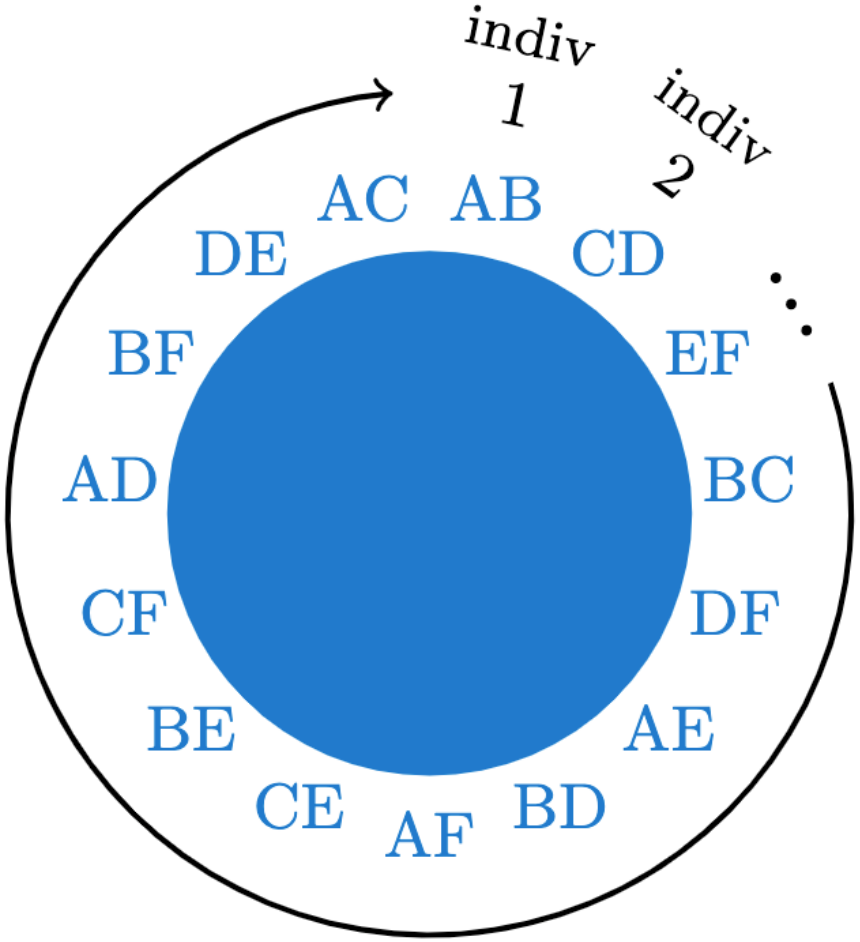
vertices : pools (A-F)

hyperedges : all possible indiv. pool assignments (ABC,ABD,...)

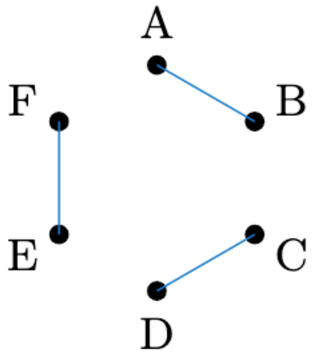
The 20 hyperedges of K_6^3 (in lexicographic order)



Another angle on the problem: hypergraph factorization

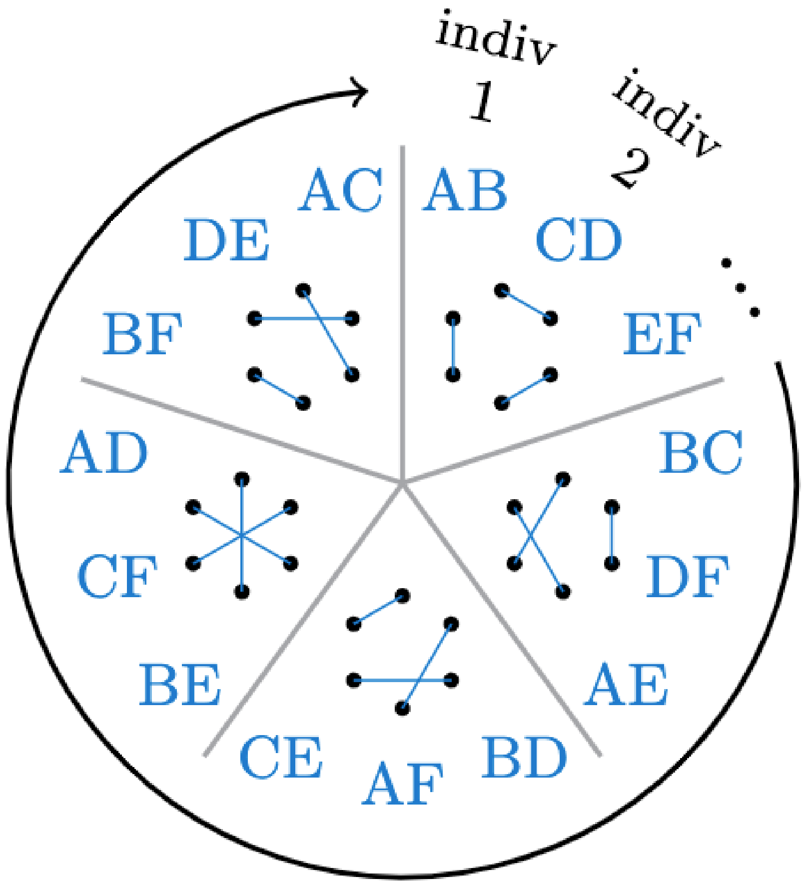


Hypergraph 1-factors

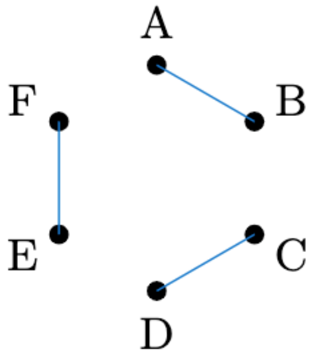


1. AB, CD, EF

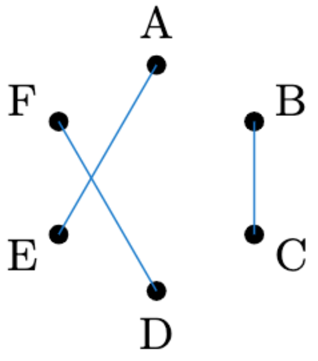
Another angle on the problem: hypergraph factorization



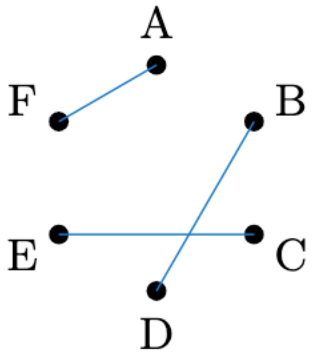
Hypergraph 1-factors



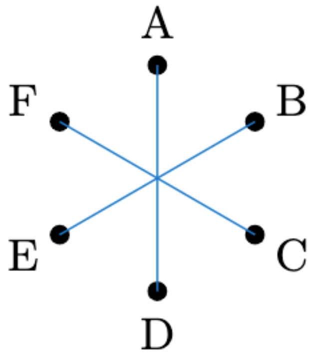
1. AB, CD, EF



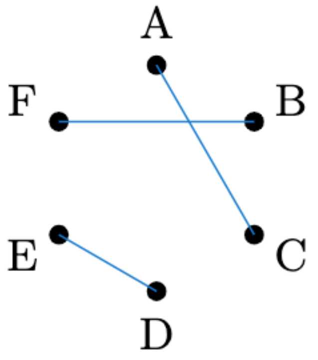
2. BC, DF, AE



3. BD, AF, CE

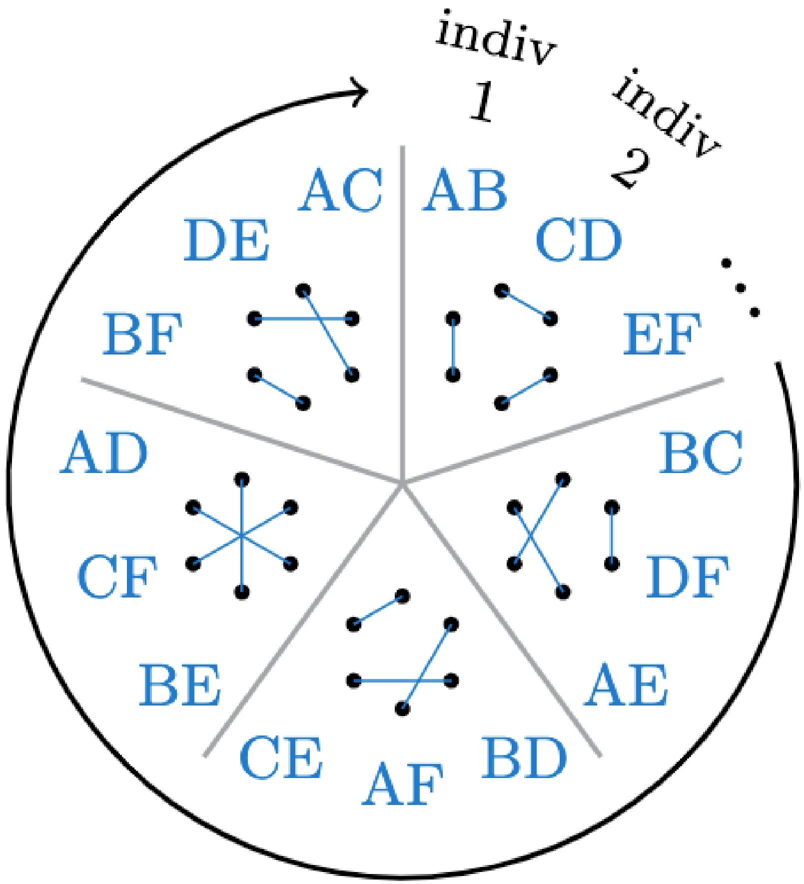


4. BE, CF, AD

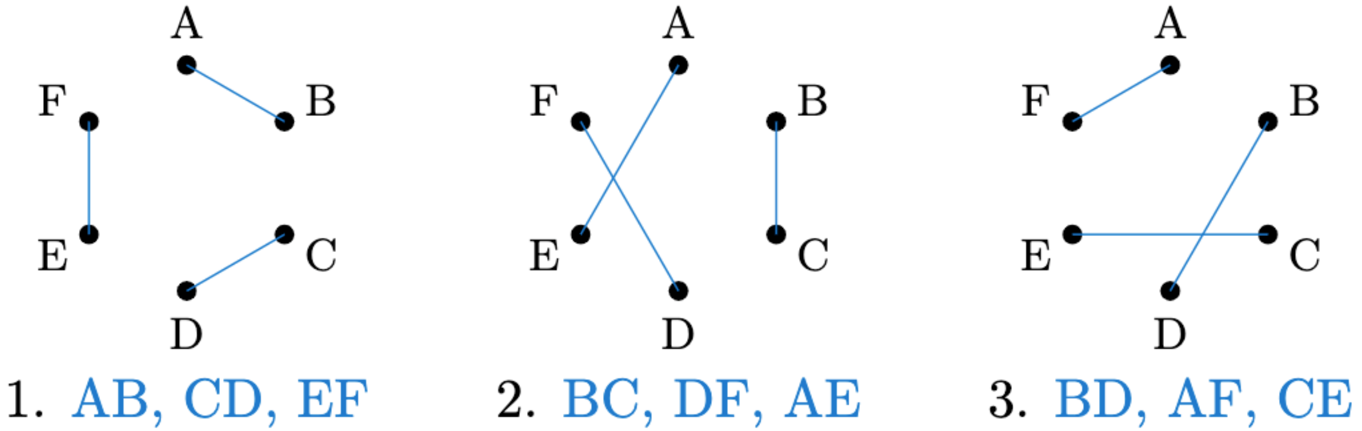


5. BF, DE, AC

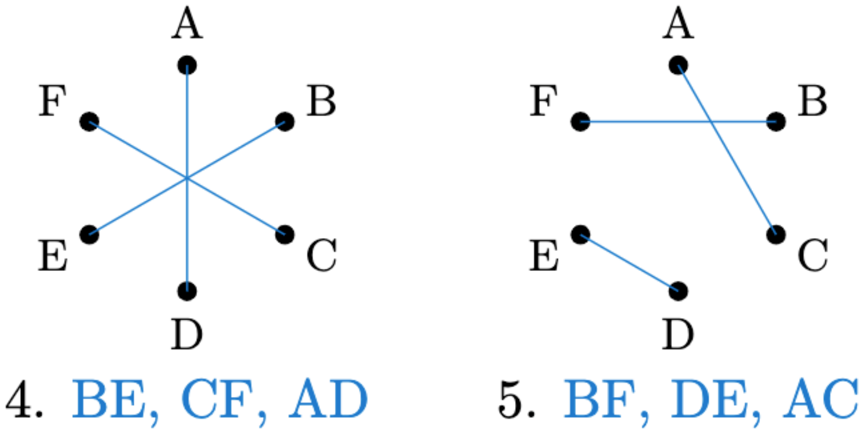
Another angle on the problem: hypergraph factorization



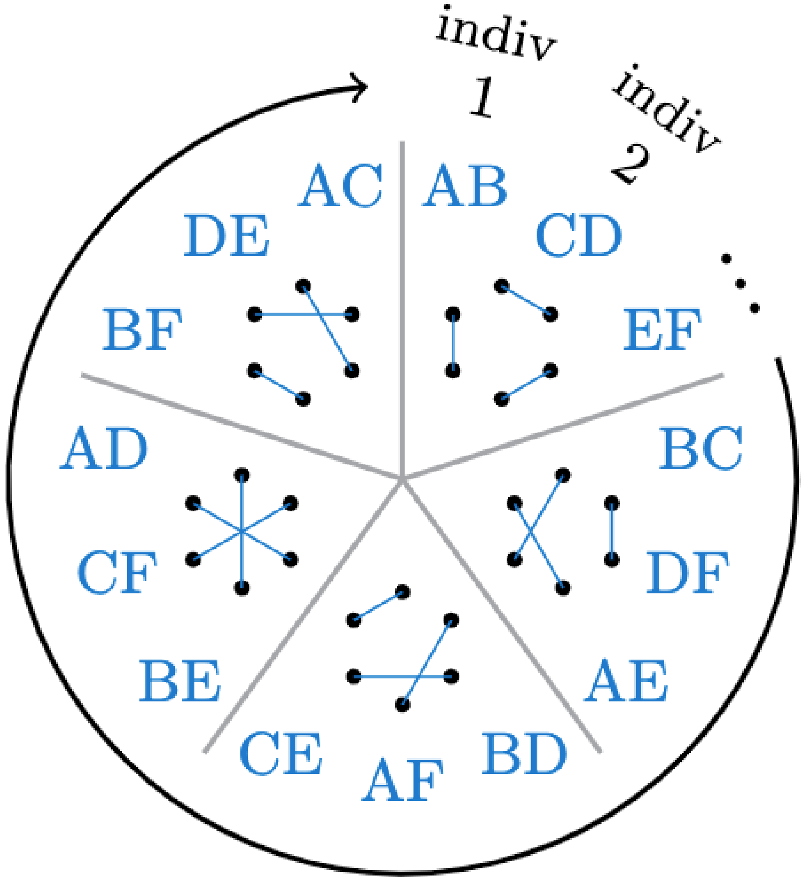
Hypergraph 1-factors



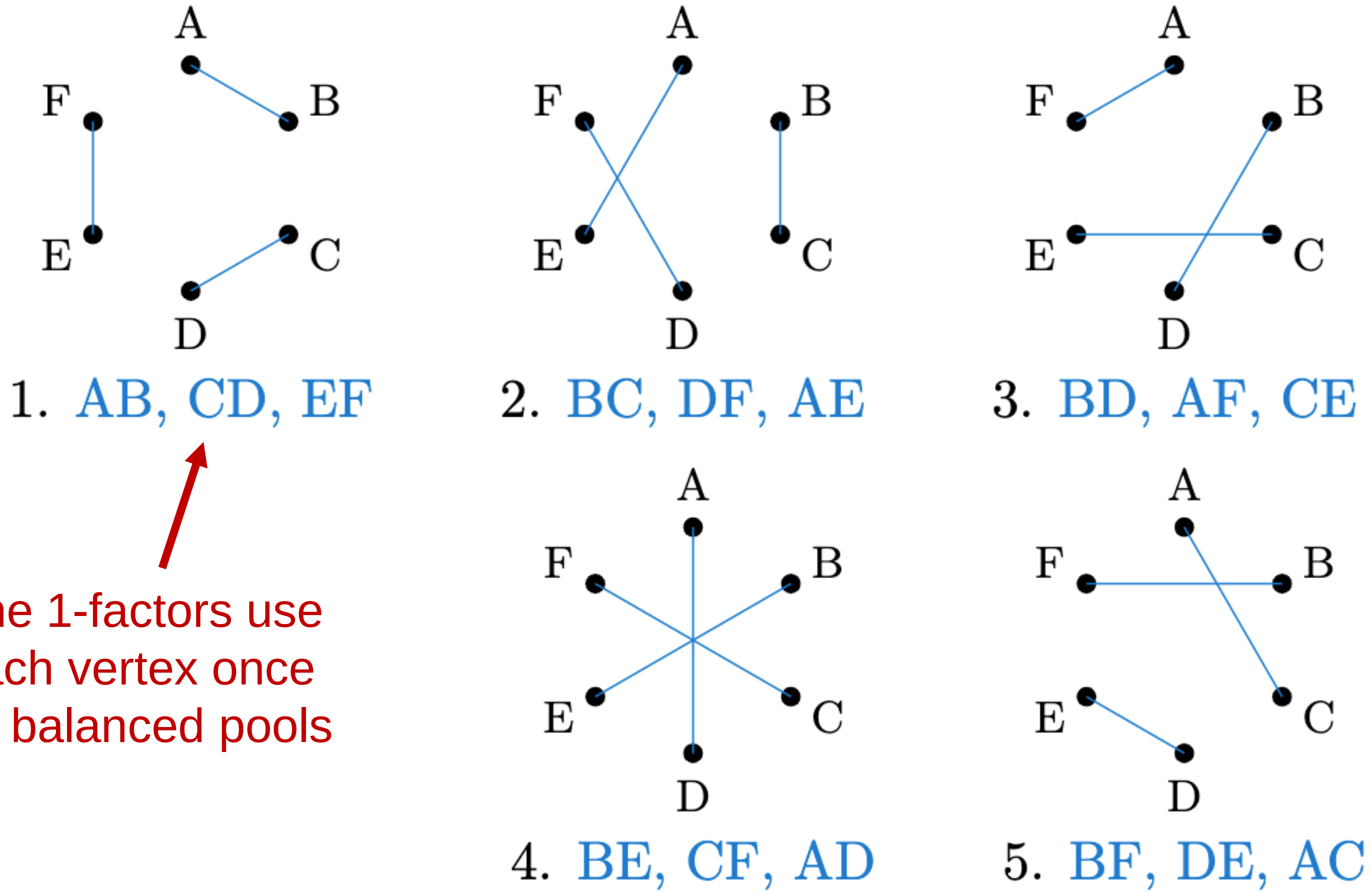
The 1-factors use each vertex once
→ balanced pools



Another angle on the problem: hypergraph factorization



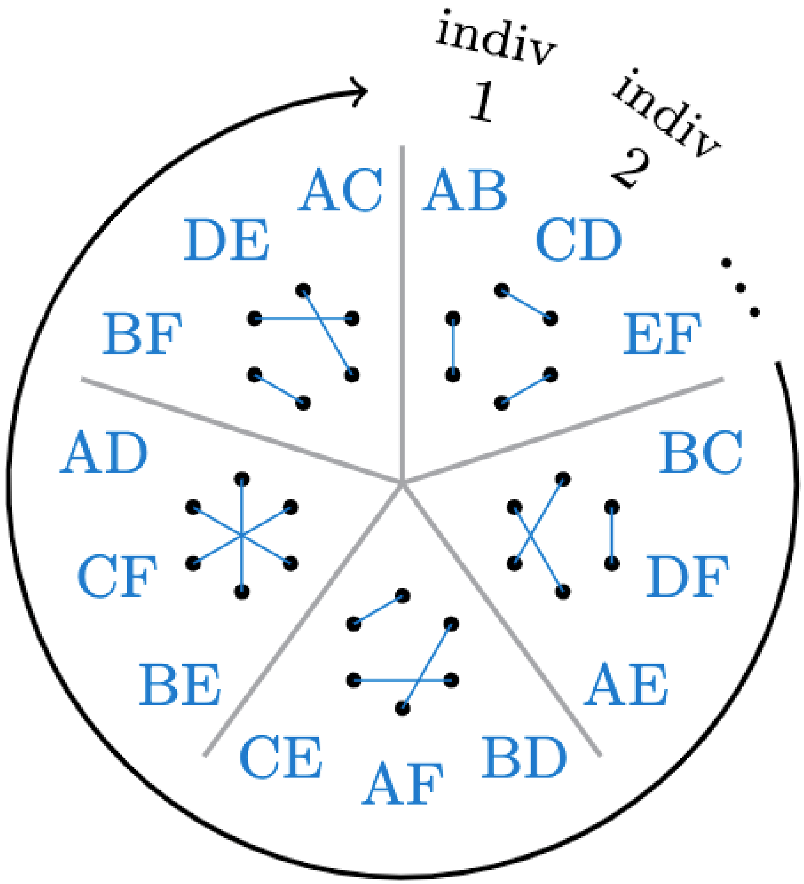
Hypergraph 1-factors



The 1-factors use each vertex once → balanced pools

The factorization uses each edge once → balanced pool combinations

Another angle on the problem: hypergraph factorization



Maximally balanced?

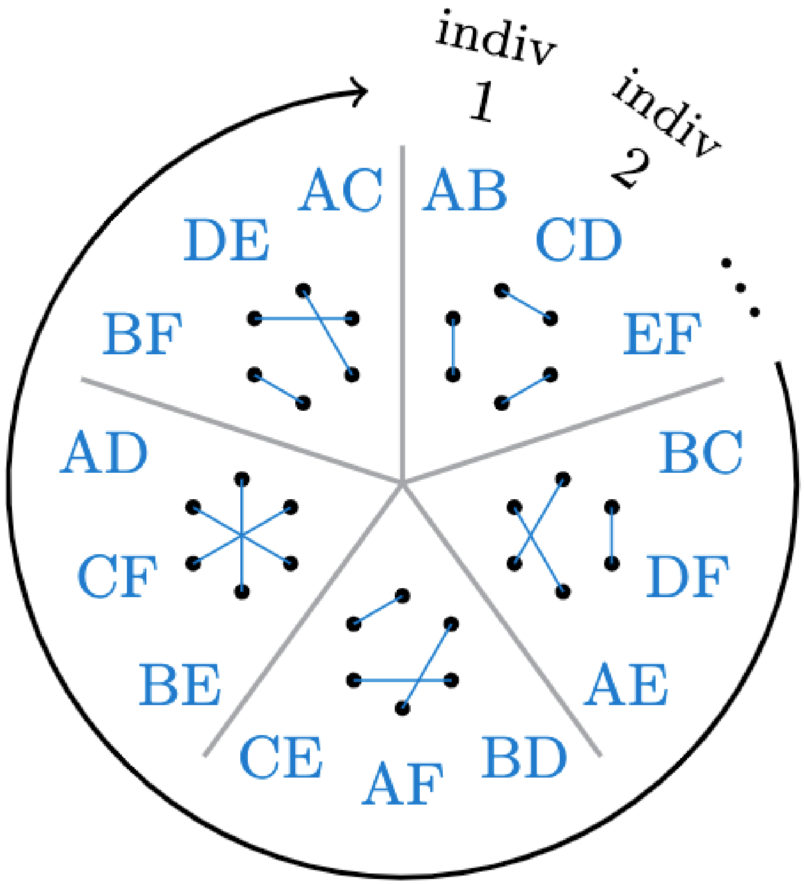
1. Assign all individuals to same number q of pools ✓

2. Assign the m pools as evenly as possible ✓

3. Assign the $\binom{m}{q}$ possible pool combinations as evenly as possible ✓

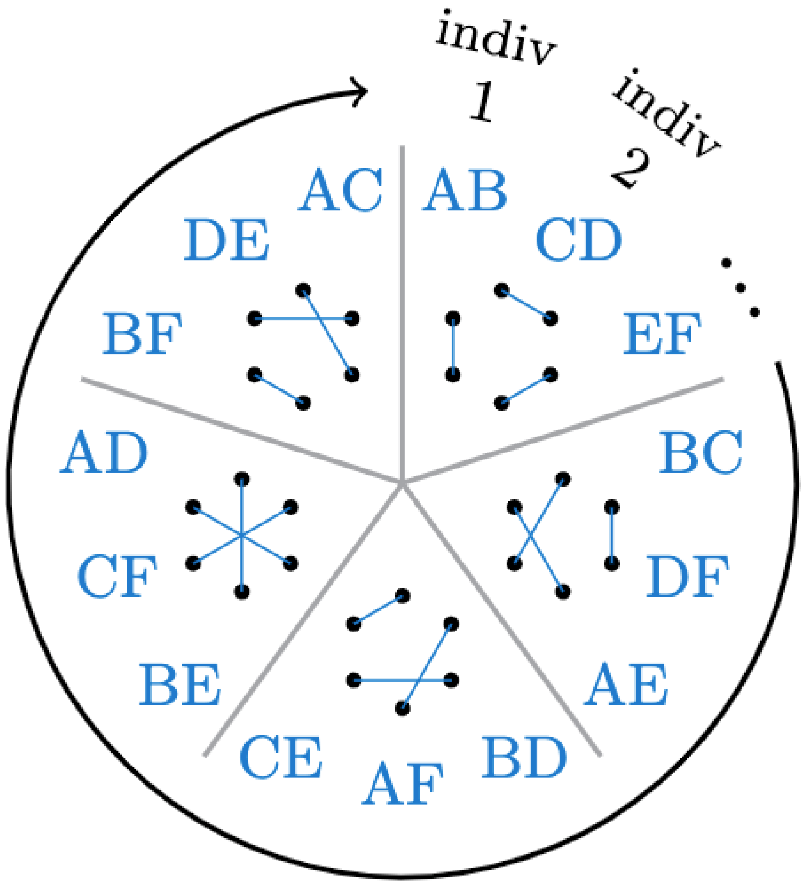
Flexible? (e.g., easy to increase # indiv in batch) ✓

Another angle on the problem: hypergraph factorization



Question: does a hypergraph factorization always exist? Is this even possible in general?

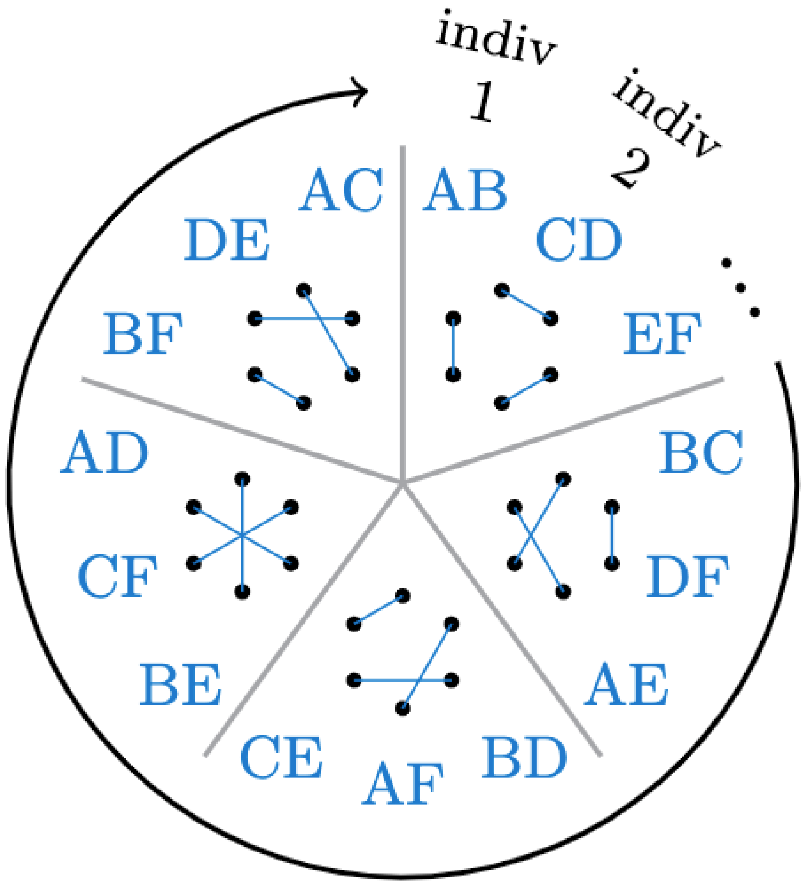
Another angle on the problem: hypergraph factorization



Question: does a hypergraph factorization always exist? Is this even possible in general?

Answer: yes! (as long as m is a multiple of q)

Another angle on the problem: hypergraph factorization



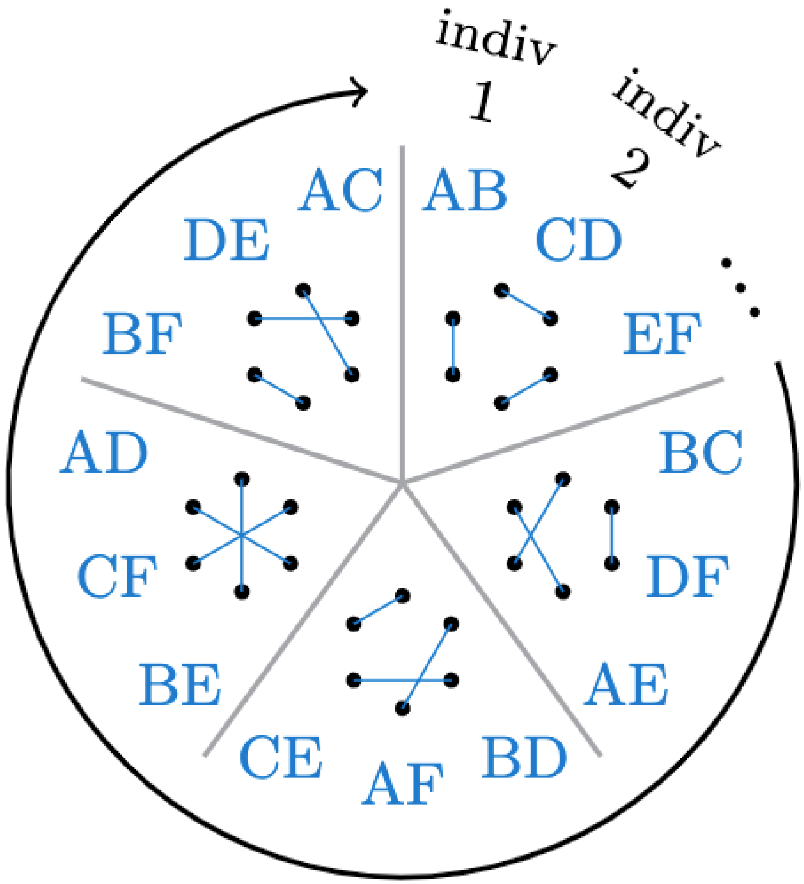
Question: does a hypergraph factorization always exist? Is this even possible in general?

Answer: yes! (as long as m is a multiple of q)

Baranyai's theorem (1972): for any $2 \leq q < m$ such that q divides m, the complete hypergraph K_m^q decomposes into 1-factors.

Well known in design theory and combinatorics, has not been used in group testing so far.

Another angle on the problem: hypergraph factorization



Question: does a hypergraph factorization always exist? Is this even possible in general?

Answer: yes! (as long as m is a multiple of q)

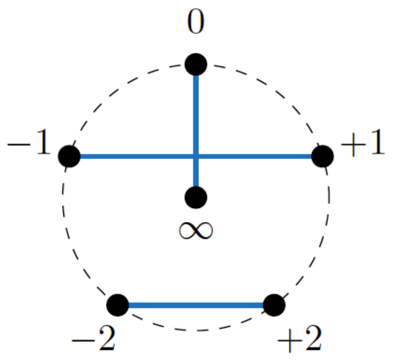
Baranyai's theorem (1972): for any $2 \leq q < m$ such that q divides m, the complete hypergraph K_m^q decomposes into 1-factors.

Well known in design theory and combinatorics, has not been used in group testing so far.

But how to efficiently construct a factorization?

Algorithm for graph factorization

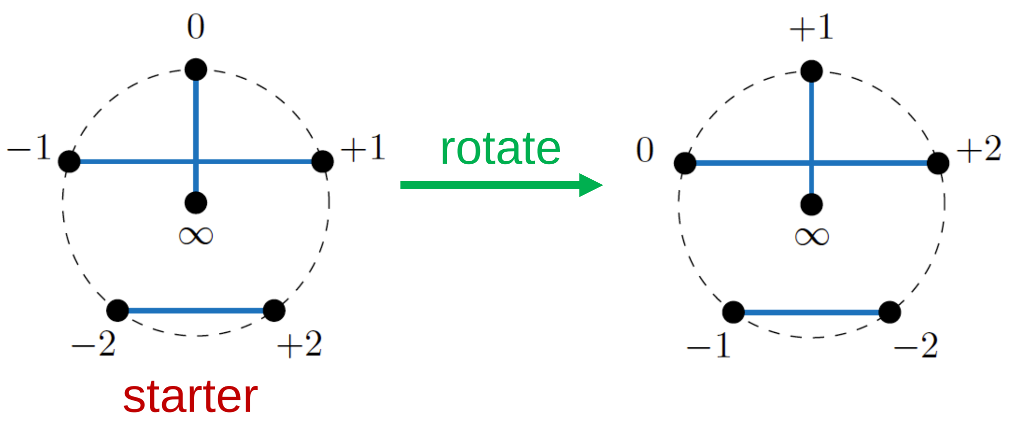
Efficient hypergraph factorization for $q = 2$ (well known)



starter

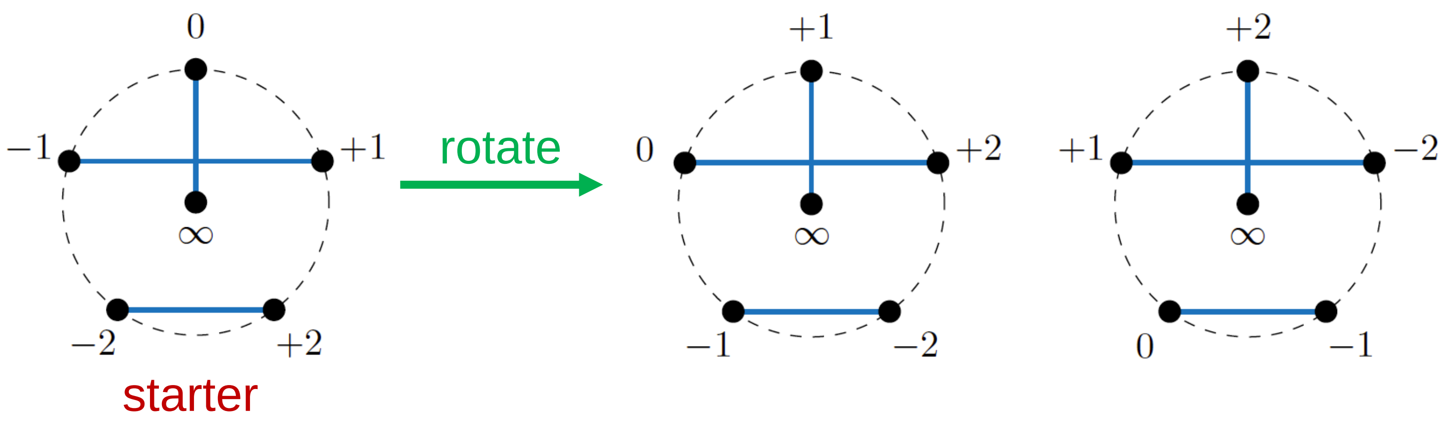
Algorithm for graph factorization

Efficient hypergraph factorization for $q = 2$



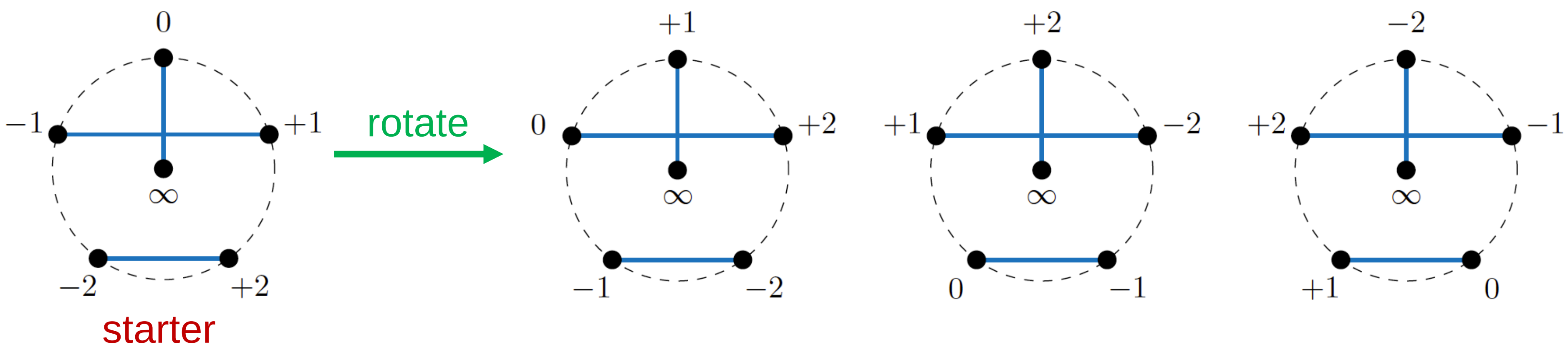
Algorithm for graph factorization

Efficient hypergraph factorization for $q = 2$



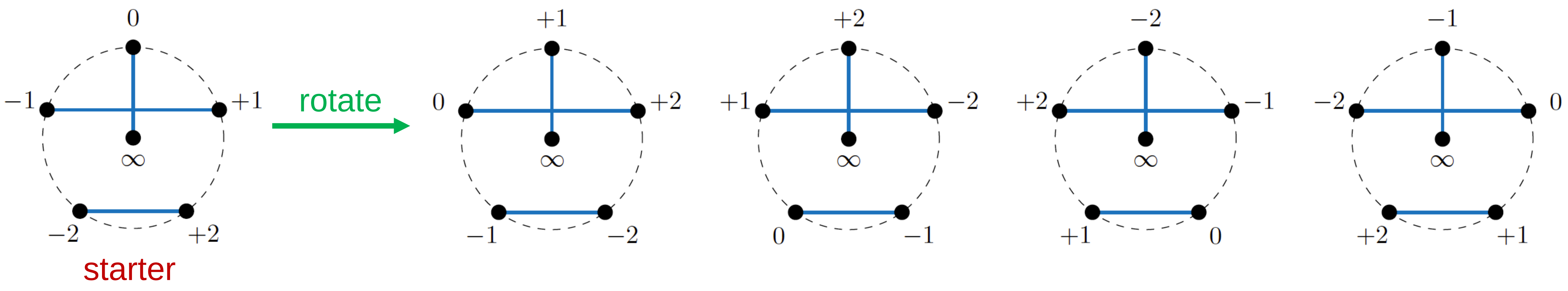
Algorithm for graph factorization

Efficient hypergraph factorization for $q = 2$



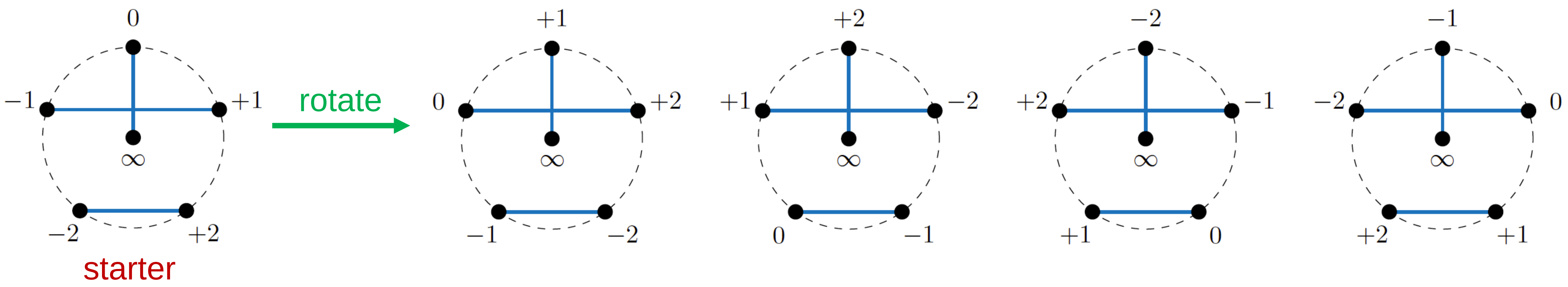
Algorithm for graph factorization

Efficient hypergraph factorization for $q = 2$

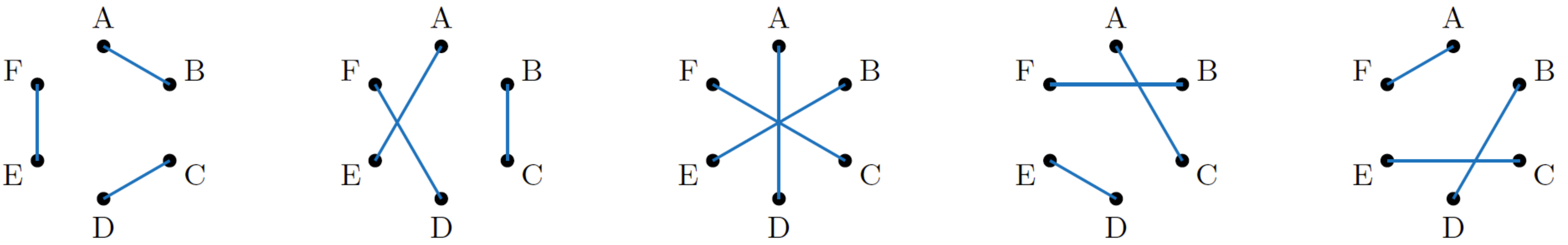


Algorithm for graph factorization

Efficient hypergraph factorization for $q = 2$



With labels $A : 0$ $B : \infty$ $C : +1$ $D : -1$ $E : +2$ $F : -2$



AB, CD, EF BC, DF, AE BE, CF, AD BF, DE, AC BD, AF, CE

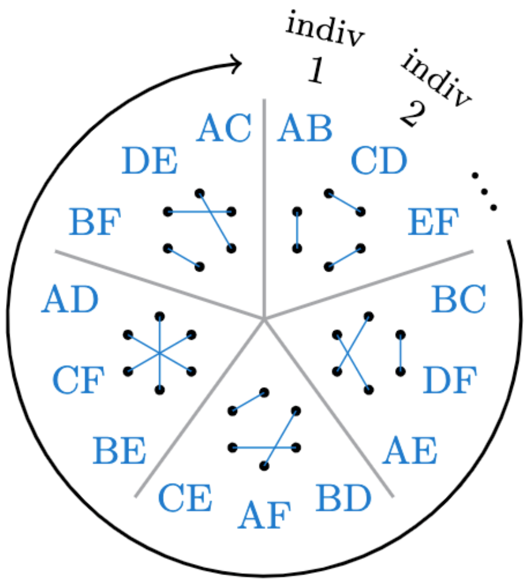
Construction for $q = 3$

- For q=3, construction due to Thomas Beth (1979)

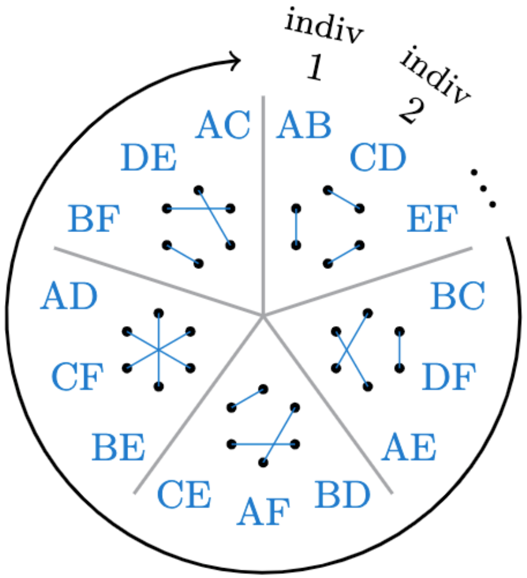
- Suppose m=6k, such that r = 6k-1 is a prime
- Consider Galois Field GF(r)
  - Adjoin infinity, to get projective line PG(1,r)
- Consider fractional linear map f: PG(1,r) → PG(1,r), f(x) = -(1+x)/x
  - Note that f is a fixed-point free map of order 3
- Let O be the partition of PG(1,r) into the (r+1)/3 orbits of f
- Let x be a primitive element of GF(r)

- Then, the partitions induced by LO + g, $L = x^j$, j = 1, … , (r-1)/2, g in GF(r) form a 1-factorization of GF(r)

How to decode the results for $q > 1$?



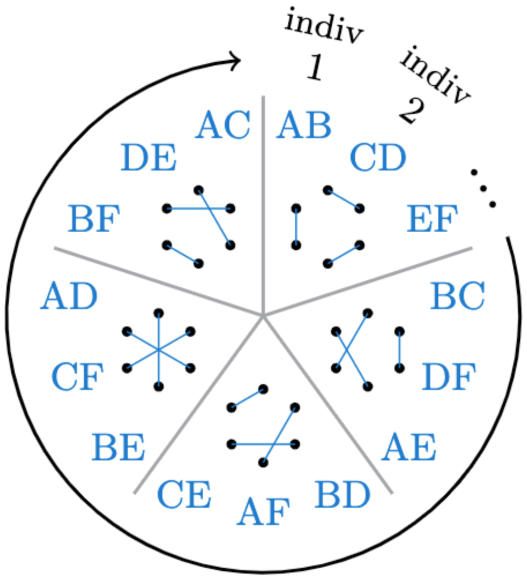
How to decode the results for $q > 1$?



Individual pool assignments from hypergraph factorization

	AB	CD	EF	BC	DF	AE	BD	AF	CE	BE	CF	AD	
Individual	1	2	3	4	5	6	7	8	9	10	11	12	Test
Pool A	●					●		●				●	
Pool B	●			●			●			●			
Pool C		●		●					●		●		
Pool D		●			●		●					●	
Pool E			●			●			●	●			
Pool F			●		●			●			●		
Decoding													

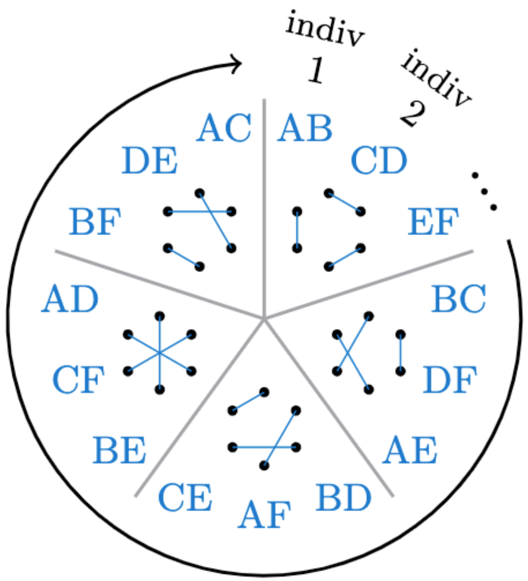
How to decode the results for $q > 1$?



Individual pool assignments from hypergraph factorization

	AB	CD	EF	BC	DF	AE	BD	AF	CE	BE	CF	AD	
Individual	1	2	3	4	5	6	7	8	9	10	11	12	Test
Pool A	●					●		●				●	⊖
Pool B	●			●			●			●			⊕
Pool C		●		●					●		●		⊕
Pool D		●			●		●					●	⊕
Pool E			●			●			●	●			⊖
Pool F			●		●			●			●		⊖
Decoding													

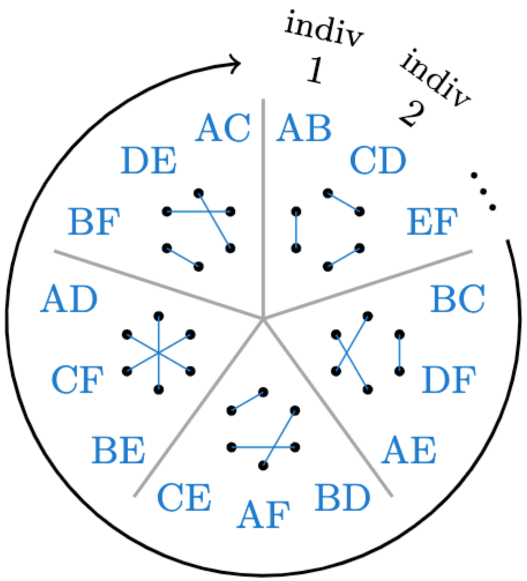
How to decode the results for $q > 1$?



Individual pool assignments from hypergraph factorization

	AB	CD	EF	BC	DF	AE	BD	AF	CE	BE	CF	AD	
Individual	1	2	3	4	5	6	7	8	9	10	11	12	Test
Pool A	●					●		●				●	⊖
Pool B	●			●			●			●			⊕
Pool C		●		●					●		●		⊕
Pool D		●			●		●					●	⊕
Pool E			●			●			●	●			⊖
Pool F			●		●			●			●		⊖
Decoding													

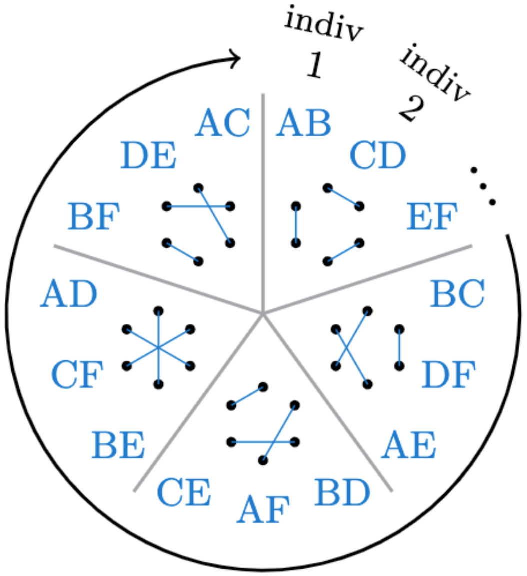
How to decode the results for $q > 1$?



Individual pool assignments from hypergraph factorization

	AB	CD	EF	BC	DF	AE	BD	AF	CE	BE	CF	AD	
Individual	1	2	3	4	5	6	7	8	9	10	11	12	Test
Pool A	●					●		●				●	⊖
Pool B	●			●			●			●			⊕
Pool C		●		●					●		●		⊕
Pool D		●			●		●					●	⊕
Pool E			●			●			●	●			⊖
Pool F			●		●			●			●		⊖
Decoding	⊖		⊖		⊖	⊖		⊖	⊖	⊖	⊖	⊖	

How to decode the results for $q > 1$?

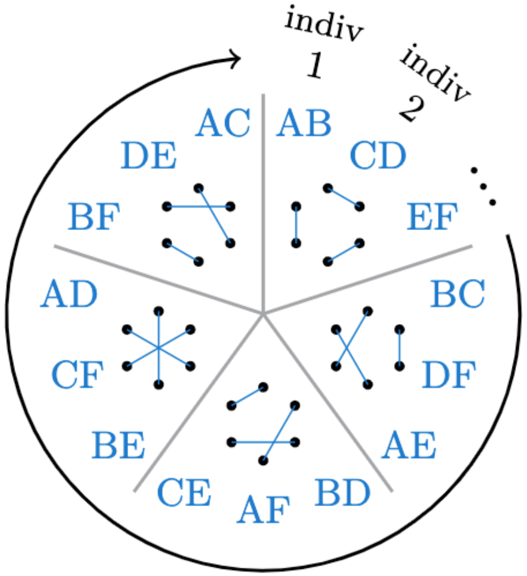


Individual pool assignments from hypergraph factorization

	AB	CD	EF	BC	DF	AE	BD	AF	CE	BE	CF	AD	
Individual	1	2	3	4	5	6	7	8	9	10	11	12	Test
Pool A	●					●		●				●	⊖
Pool B	●			●			●			●			⊕
Pool C		●		●					●		●		⊕
Pool D		●			●		●					●	⊕
Pool E			●			●			●	●			⊖
Pool F			●		●			●			●		⊖
Decoding	⊖	P	⊖	P	⊖	⊖	P	⊖	⊖	⊖	⊖	⊖	

Putative positives – to be individually tested.

How to decode the results for $q > 1$?



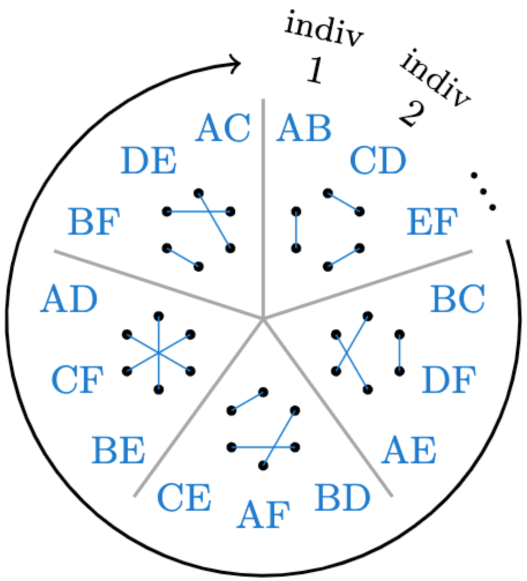
Individual pool assignments from hypergraph factorization

	AB	CD	EF	BC	DF	AE	BD	AF	CE	BE	CF	AD	
Individual	1	2	3	4	5	6	7	8	9	10	11	12	Test
Pool A	●					●		●				●	⊖
Pool B	●			●			●			●			⊕
Pool C		●		●					●		●		⊕
Pool D		●			●		●					●	⊕
Pool E			●			●			●	●			⊖
Pool F			●		●			●			●		⊖
Decoding	⊖	P	⊖	P	⊖	⊖	P	⊖	⊖	⊖	⊖	⊖	

Putative positives – to be individually tested.

Decoding is simple – can be done with pencil and paper!

How to decode the results for $q > 1$?



Individual pool assignments from hypergraph factorization

	AB	CD	EF	BC	DF	AE	BD	AF	CE	BE	CF	AD	
Individual	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**Test**
Pool A	●					●		●				●	⊖
Pool B	●			●			●			●			⊕
Pool C		●		●					●		●		⊕
Pool D		●			●		●					●	⊕
Pool E			●			●			●	●			⊖
Pool F			●		●			●			●		⊖
Decoding	⊖	P	⊖	P	⊖	⊖	P	⊖	⊖	⊖	⊖	⊖	

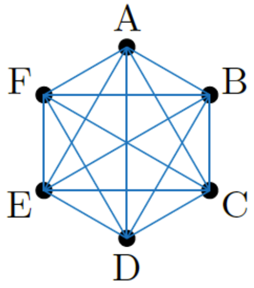
Putative positives – to be individually tested.

Decoding is simple – can be done with pencil and paper!

Note: Can also incorporate error correction by allowing putative positives to be in some negative pools.

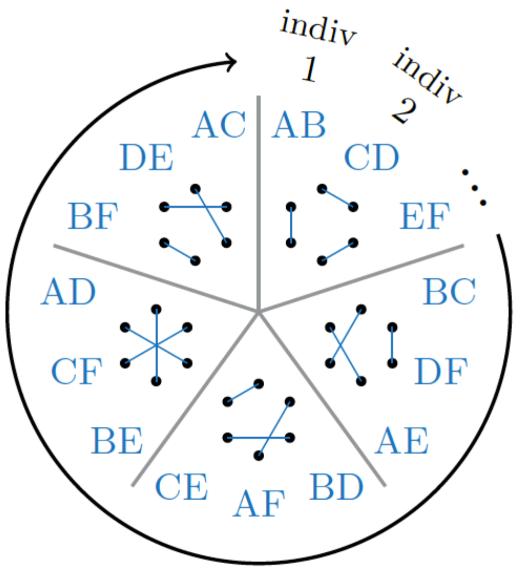
Putting it all together: HYPER

Setup: factorize the hypergraph (for $m = 6$ pools with $q = 2$ splits)



vertices : pools (A-F)
edges : all possible indiv. pool assignments (AB,AC,...)

to obtain a sequence of individual pool assignments (AB, CD, ...):



Stage 1: pooled testing of the $n = 12$ individuals.

Individual pool assignments obtained from setup

	AB	CD	EF	BC	DF	AE	BD	AF	CE	BE	CF	AD	
Individual	1	2	3	4	5	6	7	8	9	10	11	12	Test
Pool A	●					●		●				●	⊖
Pool B	●			●			●			●			⊕
Pool C		●		●					●		●		⊕
Pool D		●			●		●					●	⊕
Pool E			●			●			●	●			⊖
Pool F			●		●			●			●		⊖
Decoding	⊖	P	⊖	P	⊖	⊖	P	⊖	⊖	⊖	⊖	⊖	

Conclusion: individuals 2, 4 and 7 are putative positives (P).

Stage 2: individual testing of the putative positives (P).

Individual	2	4	7
Test Result	⊖	⊕	⊕

Final conclusion: identify 4 and 7 as positive.

Features and limitations

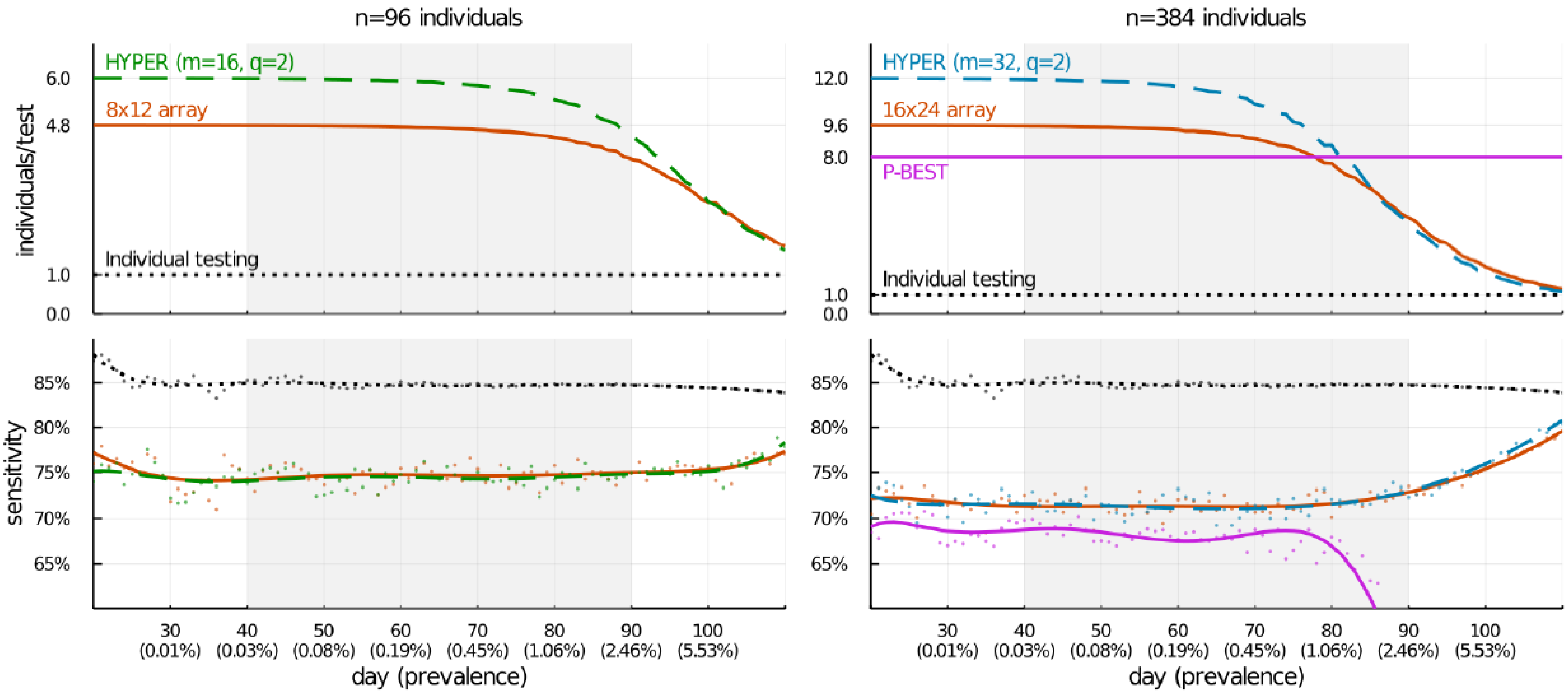
	HYPER	Random	Array designs 8×12	Array designs 16×24	P-BEST	Hypercube
# individuals per batch (n)	any	any	96	384	384	3^q/stage
# pools (m)	variable	any	20	40	48	$3q$/stage
# splits (q)	≤ 3	any	2		6	q/stage
# stages	two	two	two		one	multi
balanced pools	✓	× w.h.p.*	×		✓	✓
balanced combinations	✓	× w.h.p.*	×		✓	×
simple to implement by hand	✓	✓	✓		×	✓?
flexible / easily adapted	✓	✓	×		×	×
simple to decode by hand	✓	✓	✓		×	×
corrects false positive	✓	✓	✓		✓	✓
corrects false negatives	optional	optional	×		✓	✓?

*with high probability, i.e., probability $\gg 0$.

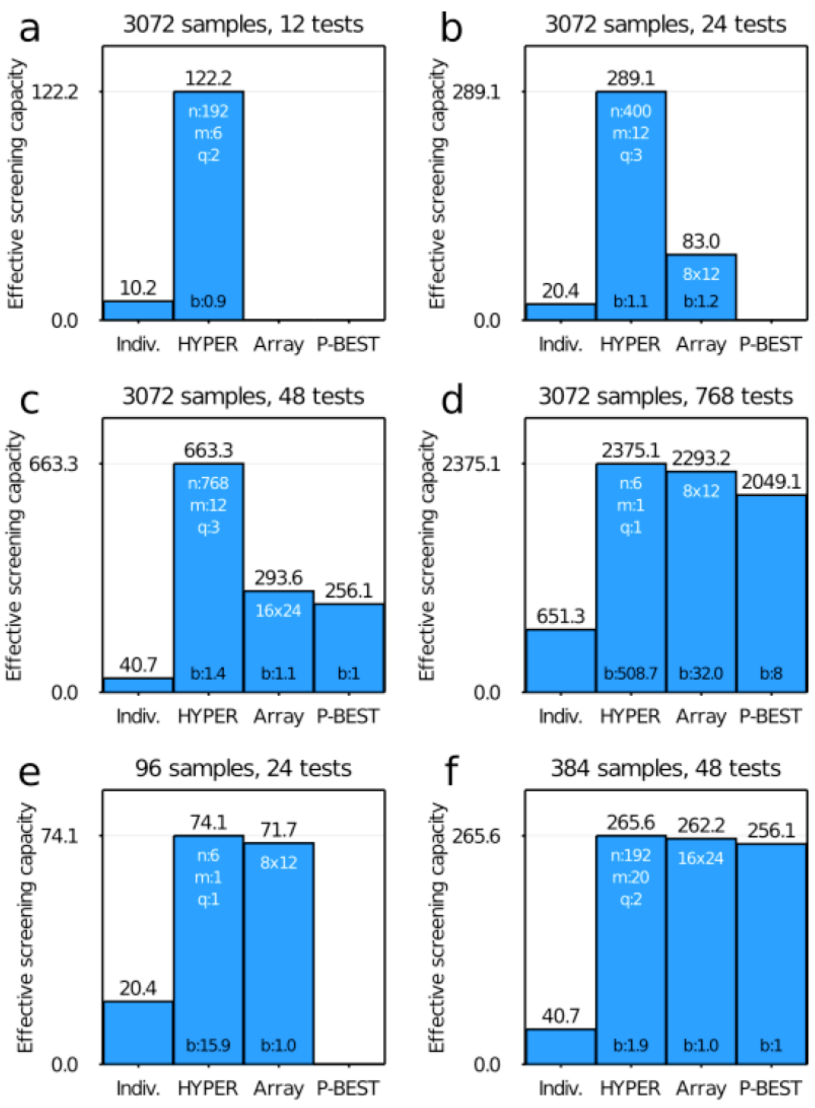
Random design: Cleary et al., 2020
Array design: Sinnott-Armstrong et al., 2020
P-BEST: Shental et al., 2020
Hypercube design: Mutesa et al., 2020

How well does it work?

Efficiency/sensitivity during simulated epidemic



How well does it work under resource constraints?



\# samples = number of samples that can be collected per day
\# tests = number of tests that can be run per day

Effective screening capacity

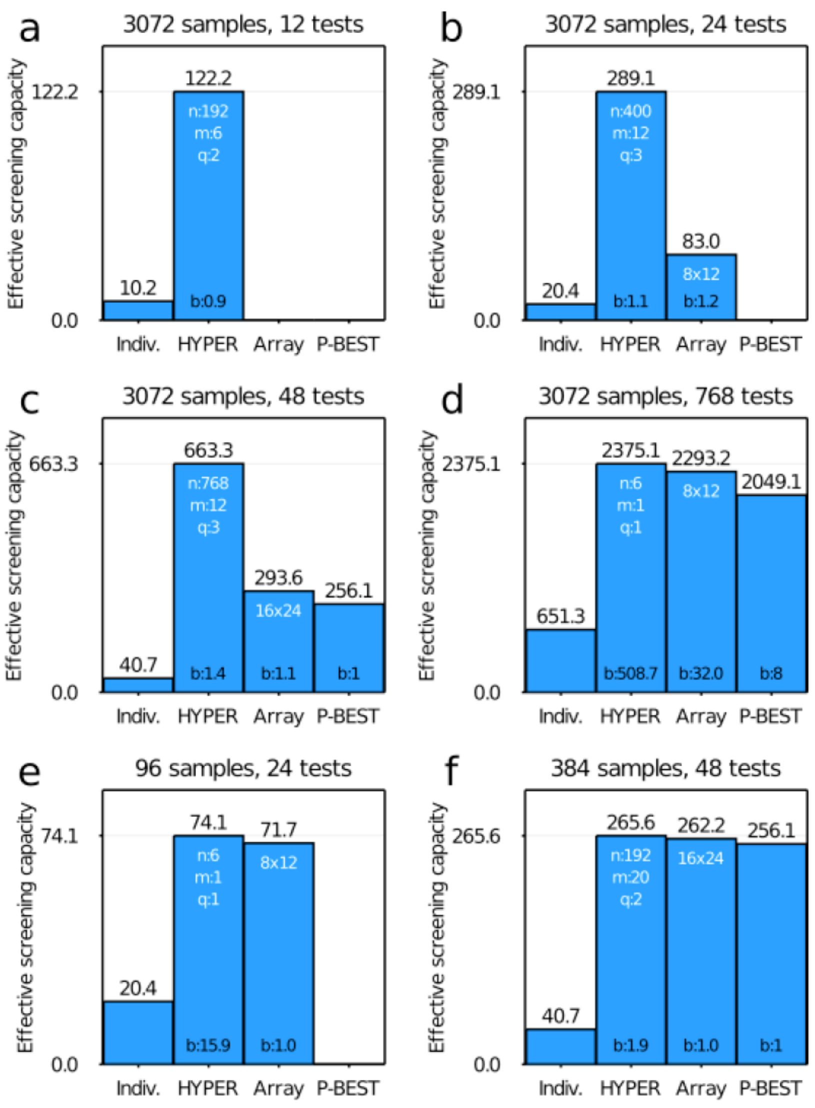
=

Expected number of positive individuals identified per day

=

(Number of individuals screened) x (Average sensitivity)

How well does it work under resource constraints?

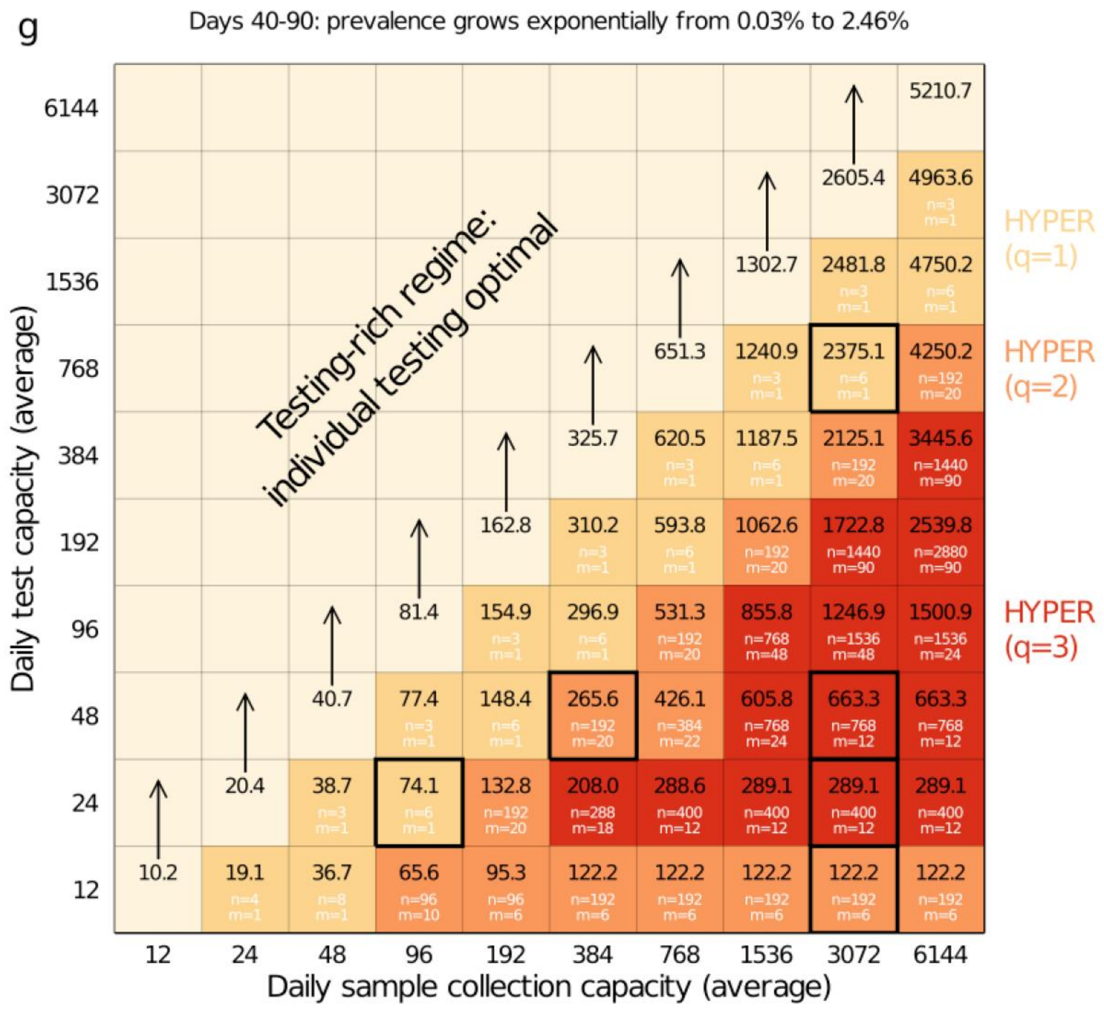


samples = number of samples that can be collected per day
tests = number of tests that can be run per day

Effective screening capacity
=
Expected number of positive individuals identified per day
=
(Number of individuals screened) x (Average sensitivity)

HYPER maximizes effective screening capacity, especially in testing scarce settings

How well does it work under resource constraints?



Black entry in each cell

=

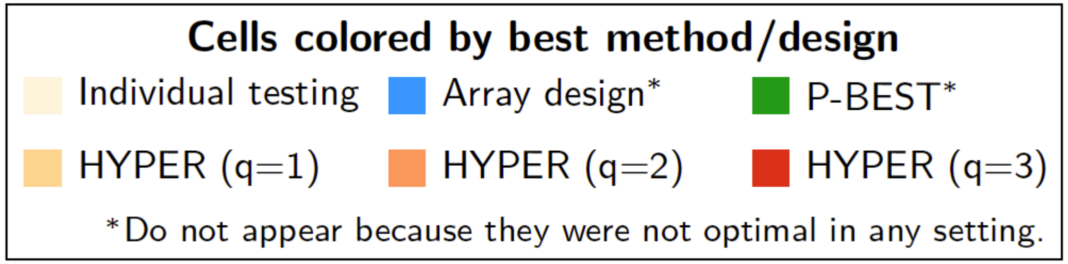
Effective screening capacity

=

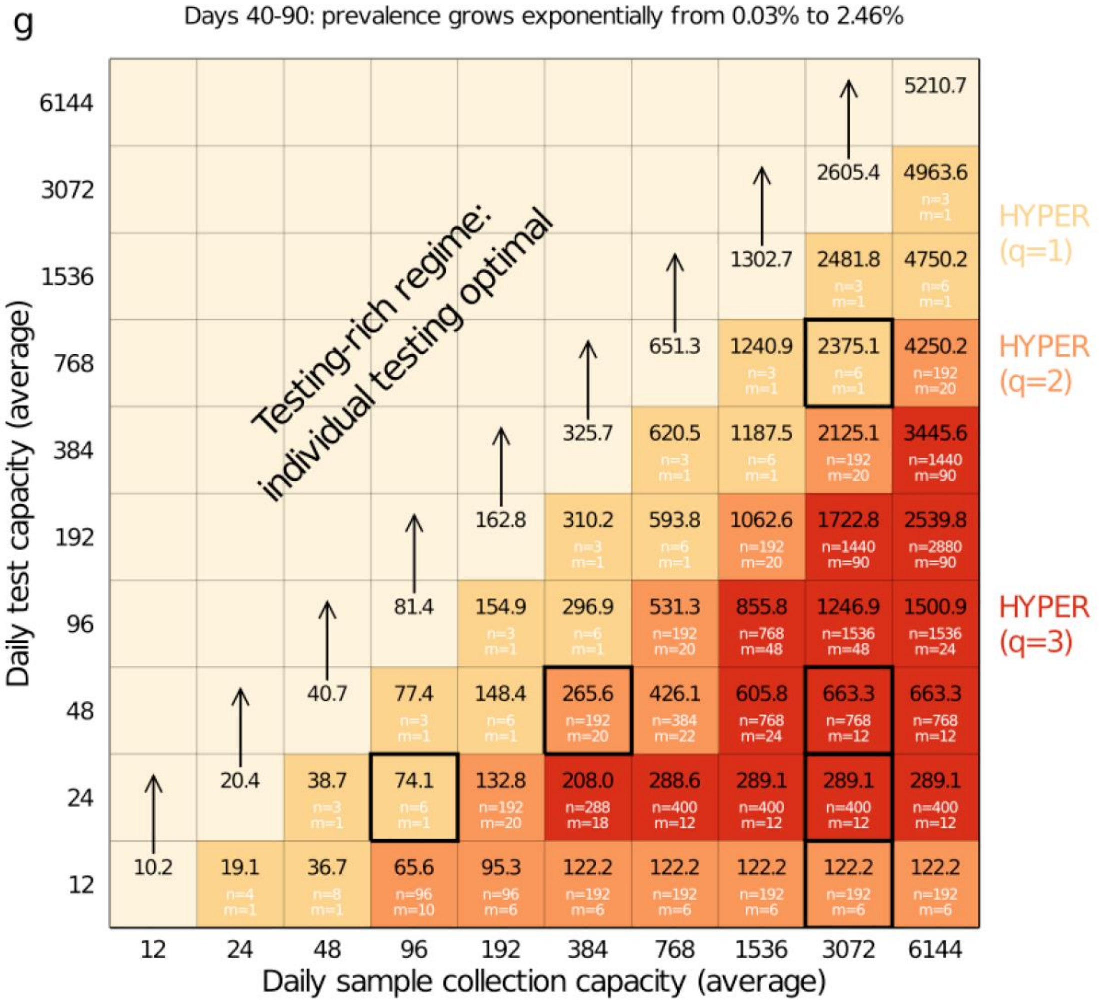
Effective number of individuals screened per day

=

Number of individuals screened x average sensitivity



How well does it work under resource constraints?



Black entry in each cell

=

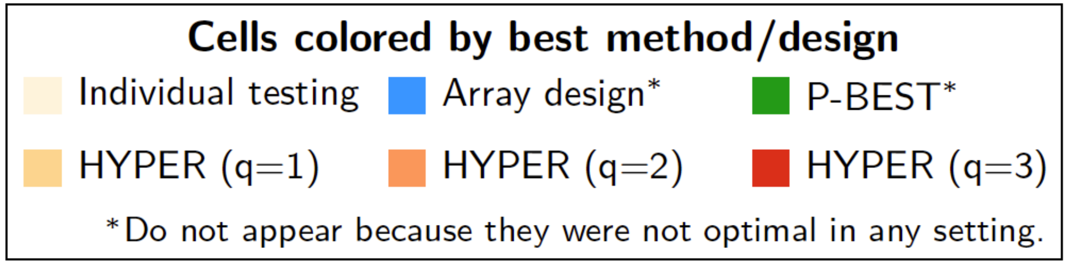
Effective screening capacity

=

Effective number of individuals screened per day

=

Number of individuals screened x average sensitivity



HYPER maximizes effective screening capacity across a range of resource constraints.

Analysis under common theoretical model

Model: n individuals, each positive independently with probability p

Analysis under common theoretical model

Model: n individuals, each positive independently with probability p

Question: What is the overall efficiency?

Supposing pool sizes are equal: $k = nq/m$ is an integer.

Analysis under common theoretical model

Model: n individuals, each positive independently with probability p

Question: What is the overall efficiency?

Supposing pool sizes are equal: $k = nq/m$ is an integer.

$$\mathbb{E}T \leq m + n \cdot \left[1 - \frac{2q}{l}(1-p)^k + \frac{q(q-1)}{l(l-1)}(1-p)^{2k-u}\right] \qquad u = \binom{m-2}{q-2} \cdot \left\lceil n/\binom{m}{q} \right\rceil$$

Analysis under common theoretical model

Model: n individuals, each positive independently with probability p

Question: What is the overall efficiency?

Supposing pool sizes are equal: $k = nq/m$ is an integer.

$$\mathbb{E}T \leq m + n \cdot \left[1 - \frac{2q}{l}(1-p)^k + \frac{q(q-1)}{l(l-1)}(1-p)^{2k-u}\right] \qquad u = \binom{m-2}{q-2} \cdot \lceil n / \binom{m}{q} \rceil$$

Equality when q=2.
More complicated when tests can be noisy…see the paper!

Analysis under common theoretical model

$$\mathbb{E}T \leq m + n \cdot \left[1 - \frac{2q}{l}(1-p)^k + \frac{q(q-1)}{l(l-1)}(1-p)^{2k-u}\right] \qquad u = \binom{m-2}{q-2} \cdot \left\lceil n \middle/ \binom{m}{q} \right\rceil$$

Analysis under common theoretical model

$$\mathbb{E}T \leq m + n \cdot \left[1 - \frac{2q}{l}(1-p)^k + \frac{q(q-1)}{l(l-1)}(1-p)^{2k-u}\right] \qquad u = \binom{m-2}{q-2} \cdot \left\lceil n \Big/ \binom{m}{q} \right\rceil$$

How is this obtained?

Analysis under common theoretical model

$$\mathbb{E}T \leq m + n \cdot \left[1 - \frac{2q}{l}(1-p)^k + \frac{q(q-1)}{l(l-1)}(1-p)^{2k-u}\right] \qquad u = \binom{m-2}{q-2} \cdot \left\lceil n \Big/ \binom{m}{q} \right\rceil$$

How is this obtained?

$$\mathbb{E}(\text{number of tests used}) = m + \sum_{i=1}^{n} \Pr(\text{individual } i \text{ is retested})$$

Analysis under common theoretical model

$$\mathbb{E}T \leq m + n \cdot \left[1 - \frac{2q}{l}(1-p)^k + \frac{q(q-1)}{l(l-1)}(1-p)^{2k-u}\right] \qquad u = \binom{m-2}{q-2} \cdot \lceil n / \binom{m}{q} \rceil$$

How is this obtained?

$$\mathbb{E}(\text{number of tests used}) = m + \sum_{i=1}^{n} \Pr(\text{individual } i \text{ is retested})$$

$$\Pr(\text{individual } i \text{ is retested}) = 1 - \Pr(\text{no positives in pools including individual } i)$$

Analysis under common theoretical model

$$\mathbb{E}T \le m + n \cdot \left[1 - \frac{2q}{l}(1-p)^k + \frac{q(q-1)}{l(l-1)}(1-p)^{2k-u}\right] \qquad u = \binom{m-2}{q-2} \cdot \left\lceil n / \binom{m}{q} \right\rceil$$

How is this obtained?

$$\mathbb{E}(\text{number of tests used}) = m + \sum_{i=1}^{n} \Pr(\text{individual } i \text{ is retested})$$

$$\Pr(\text{individual } i \text{ is retested}) = 1 - \Pr(\text{no positives in pools including individual } i)$$

Task is to bound this probability!

Analysis under common theoretical model

$$\mathbb{E}T \leq m + n \cdot \left[1 - \frac{2q}{l}(1-p)^k + \frac{q(q-1)}{l(l-1)}(1-p)^{2k-u}\right] \qquad u = \binom{m-2}{q-2} \cdot \left\lceil n \Big/ \binom{m}{q} \right\rceil$$

How is this obtained?

$$\mathbb{E}(\text{number of tests used}) = m + \sum_{i=1}^{n} \Pr(\text{individual } i \text{ is retested})$$

$$\Pr(\text{individual } i \text{ is retested}) = 1 - \Pr(\text{no positives in pools including individual } i)$$

Task is to bound this probability!

Bonferroni inequality / Dawson-Sankoff inequality
+ bound on pool overlaps ← balance comes up here!

Analysis under common theoretical model

Question: Optimal number of pools for $q = 2$ at low prevalence?
(in the regime of large n, i.e., big batches)

Analysis under common theoretical model

Question: Optimal number of pools for $q = 2$ at low prevalence?
(in the regime of large n, i.e., big batches)

Taking $n \to \infty$, with $y = m/n$

$$E = \frac{T}{n} = \frac{m}{n} + 1 - 2(1-p)^{2n/m} + (1-p)^{4n/m-1} = y + 1 - 2(1-p)^{2/y} + (1-p)^{4/y-1}.$$

Analysis under common theoretical model

Question: Optimal number of pools for $q = 2$ at low prevalence?
(in the regime of large n, i.e., big batches)

Taking $n \to \infty$, with $y = m/n$

$$E = \frac{T}{n} = \frac{m}{n} + 1 - 2(1-p)^{2n/m} + (1-p)^{4n/m-1} = y + 1 - 2(1-p)^{2/y} + (1-p)^{4/y-1}.$$

Set diff equal to zero and approximate solve in low prevalence limit:

$$\frac{\partial E}{\partial y} = 1 + \frac{4(1-p)^{2/y}\log(1-p)}{y^2} - \frac{4(1-p)^{4/y-1}\log(1-p)}{y^2}.$$

Analysis under common theoretical model

Question: Optimal number of pools for $q = 2$ at low prevalence?
(in the regime of large n, i.e., big batches)

Taking $n \to \infty$, with $y = m/n$

$$E = \frac{T}{n} = \frac{m}{n} + 1 - 2(1-p)^{2n/m} + (1-p)^{4n/m-1} = y + 1 - 2(1-p)^{2/y} + (1-p)^{4/y-1}.$$

Set diff equal to zero and approximate solve in low prevalence limit:

$$\frac{\partial E}{\partial y} = 1 + \frac{4(1-p)^{2/y}\log(1-p)}{y^2} - \frac{4(1-p)^{4/y-1}\log(1-p)}{y^2}.$$

$$\xrightarrow{\text{Puiseux expansion around } p = 0 \text{ (of Taylor approx.)}} \quad y^\star \approx 2p^{2/3} - p.$$

Analysis under common theoretical model

Question: Optimal number of pools for $q = 2$ at low prevalence?
(in the regime of large n, i.e., big batches)

Taking $n \to \infty$, with $y = m/n$

$$E = \frac{T}{n} = \frac{m}{n} + 1 - 2(1-p)^{2n/m} + (1-p)^{4n/m-1} = y + 1 - 2(1-p)^{2/y} + (1-p)^{4/y-1}.$$

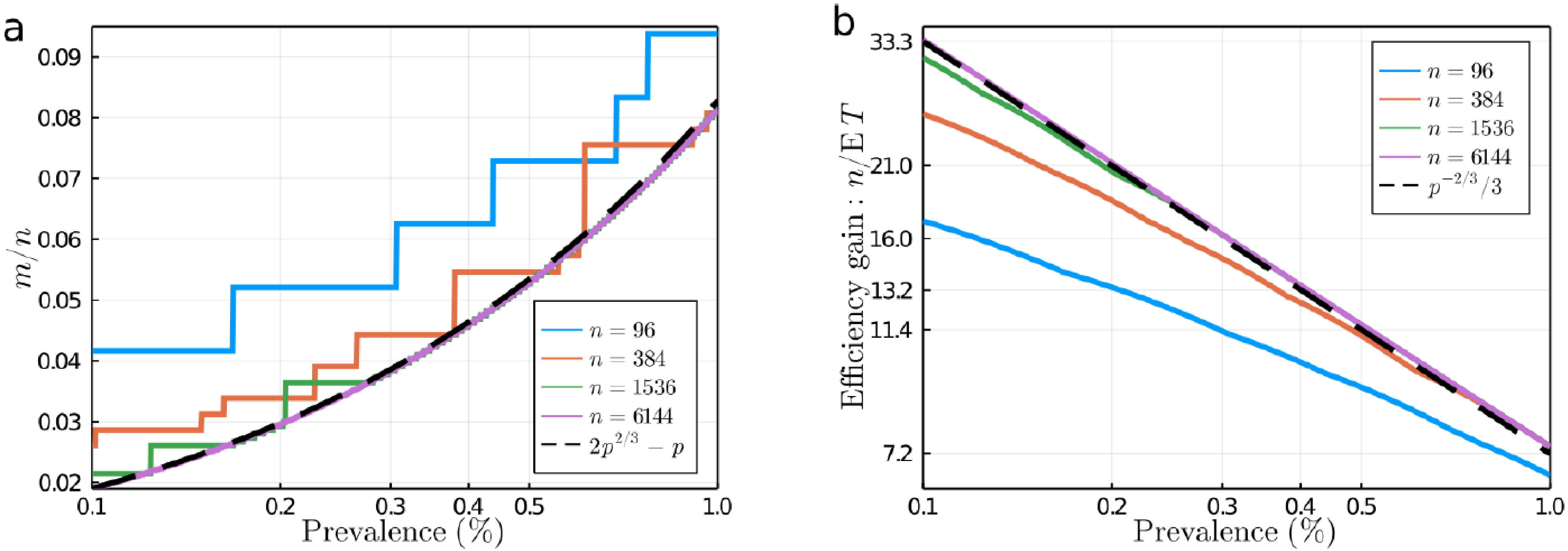
Set diff equal to zero and approximate solve in low prevalence limit:

$$\frac{\partial E}{\partial y} = 1 + \frac{4(1-p)^{2/y}\log(1-p)}{y^2} - \frac{4(1-p)^{4/y-1}\log(1-p)}{y^2}.$$

$$\xrightarrow{\text{Puiseux expansion around } p = 0 \text{ (of Taylor approx.)}} \quad y^\star \approx 2p^{2/3} - p.$$

Dorfman efficiency: $2p^{1/2}$ Corresponding efficiency: $\mathbb{E}(T)/n \approx 3p^{2/3}$

Analysis under common theoretical model



Numerical optimization vs. asymptotic approximation

Conclusion

Today we saw:

- Pooled testing to increase screening capacity given limited testing resources
- Finding balanced designs for $q > 1$ via hypergraph factorization → HYPER
- Efficient construction of hypergraph factorization for $q = 2{,}3$
- Performance of HYPER under a realistic simulation
- Analysis under common theoretical model (noiseless case)

Check out our online tool: http://hyper.covid19-analysis.org

See paper (https://doi.org/10.1101/2021.02.24.21252394) for:

- More simulation studies
- Analysis under theoretical model with noisy tests
- …

Conclusion

Today we saw:

- Pooled testing to increase screening capacity given limited testing resources
- Finding balanced designs for $q > 1$ via hypergraph factorization → HYPER
- Efficient construction of hypergraph factorization for $q = 2,3$
- Performance of HYPER under a realistic simulation
- Analysis under common theoretical model (noiseless case)

Check out our online tool: http://hyper.covid19-analysis.org

See paper (https://doi.org/10.1101/2021.02.24.21252394) for:

- More simulation studies
- Analysis under theoretical model with noisy tests
- …

Thanks!