Combinatorial group testing on hypergraphs

Thaís Bardini Idalino
Universidade Federal de Santa Catarina - Brazil

Ottawa Mathematics and Statistics Conference
May 2025

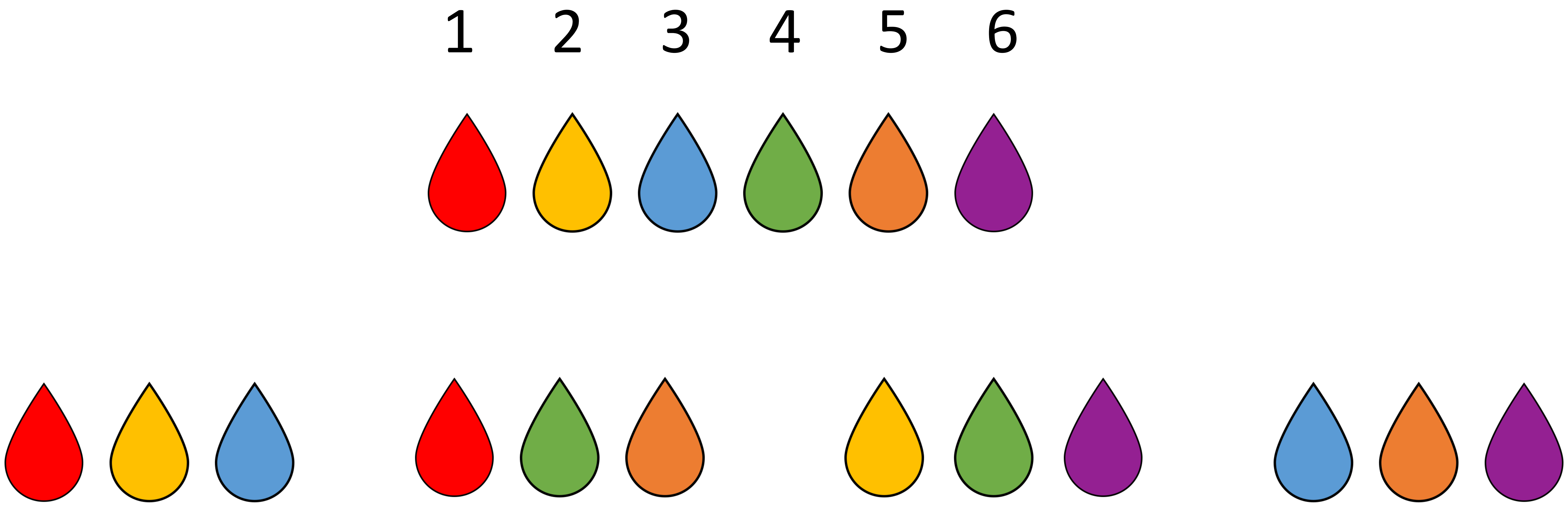
uOttawa

UFSC

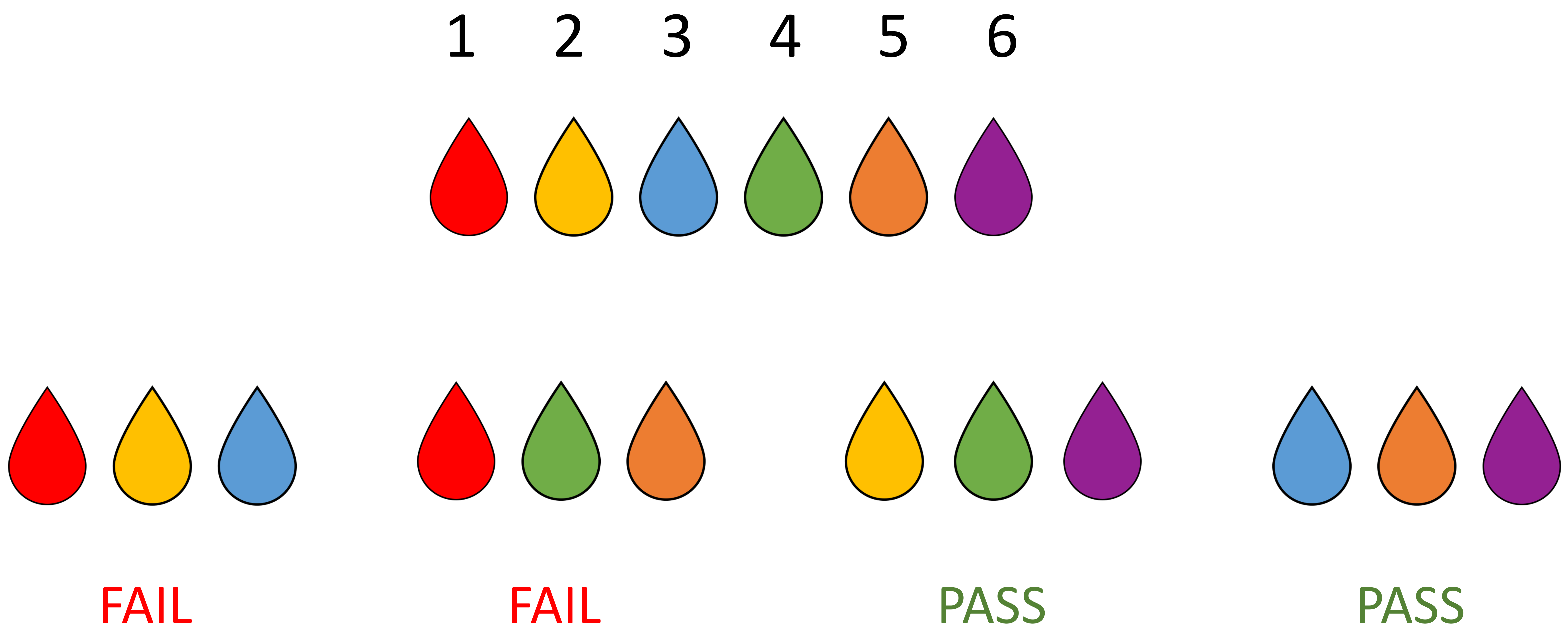
Joint work with Lucia Moura


UFSC
Undergrad
2009
2013
Masters
UFSC
uOttawa
PhD
2015
2019
Postdoc
SFU SIMON FRASER UNIVERSITY
UFSC
Professor
2021

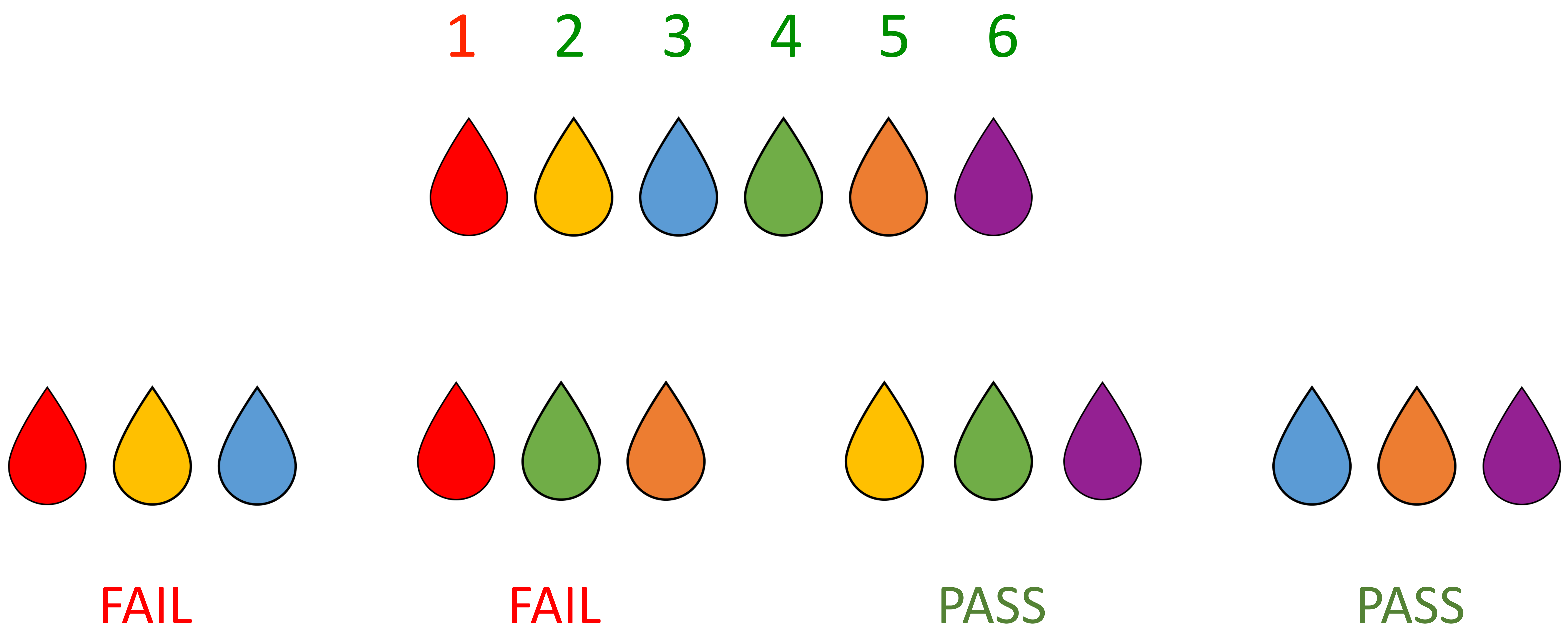
Combinatorial Group Testing



Combinatorial Group Testing



Combinatorial Group Testing



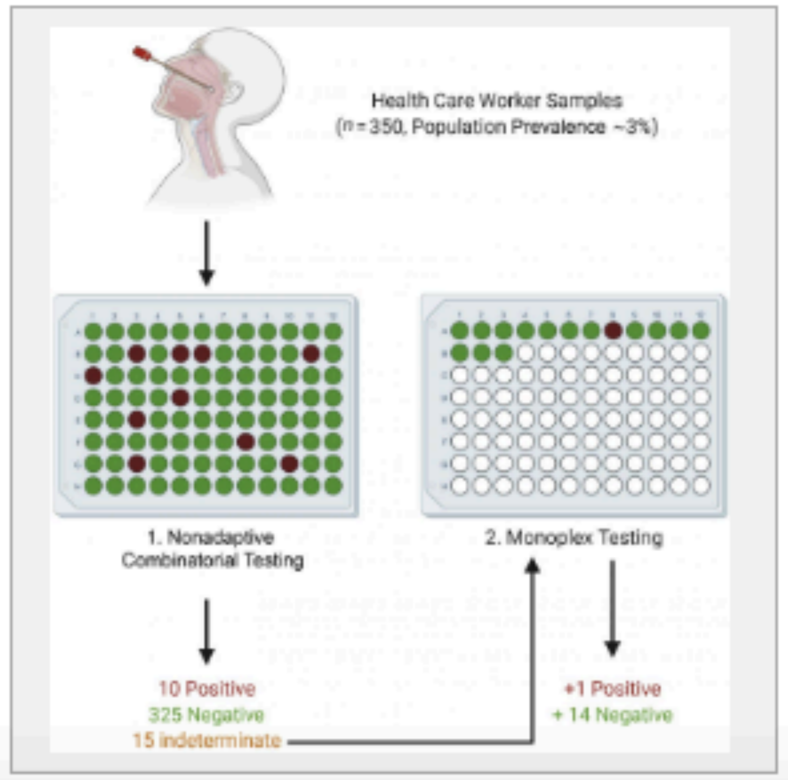
New testing strategy can speed up COVID-19 test results for healthcare workers

In The Journal of Molecular Diagnostics investigators share a new methodology for testing pooled samples that maximizes the proportion of samples resolved after a single round of testing

Peer-Reviewed Publication

ELSEVIER

Philadelphia, April 26, 2021 - Fast turnaround of COVID-19 test results for healthcare workers is critical. Investigators have now developed a COVID-19 testing strategy that maximizes the proportion of negative results after a single round of testing, allowing prompt notification of results. The method also reduces the need for increasingly limited test reagents, as fewer additional tests are required. Their strategy is described in *The Journal of Molecular Diagnostics*, published by Elsevier.



nature › scientific reports › articles › article

Article | Open access | Published: 26 July 2023

Adaptive group testing strategy for infectious diseases using social contact graph partitions

Jingyi Zhang & Lenwood S. Heath

Scientific Reports 13, Article number: 12102 (2023) | Cite this article

The Journal of Molecular Diagnostics, Vol. 23, No. 5, May 2021

ELSEVIER

the Journal of Molecular Diagnostics

jmdjournal.org

Check for updates

A Nonadaptive Combinatorial Group Testing Strategy to Facilitate Health Care Worker Screening during the Severe Acute Respiratory Syndrome Coronavirus-2 (SARS-CoV-2) Outbreak

John H. McDermott,*† Duncan Stoddard,‡ Peter J. Woolf,§ Jamie M. Ellingford,*† David Gokhale,*† Algy Taylor,* Leigh A.M. Demain,* William G. Newman,*† and Graeme Black*†

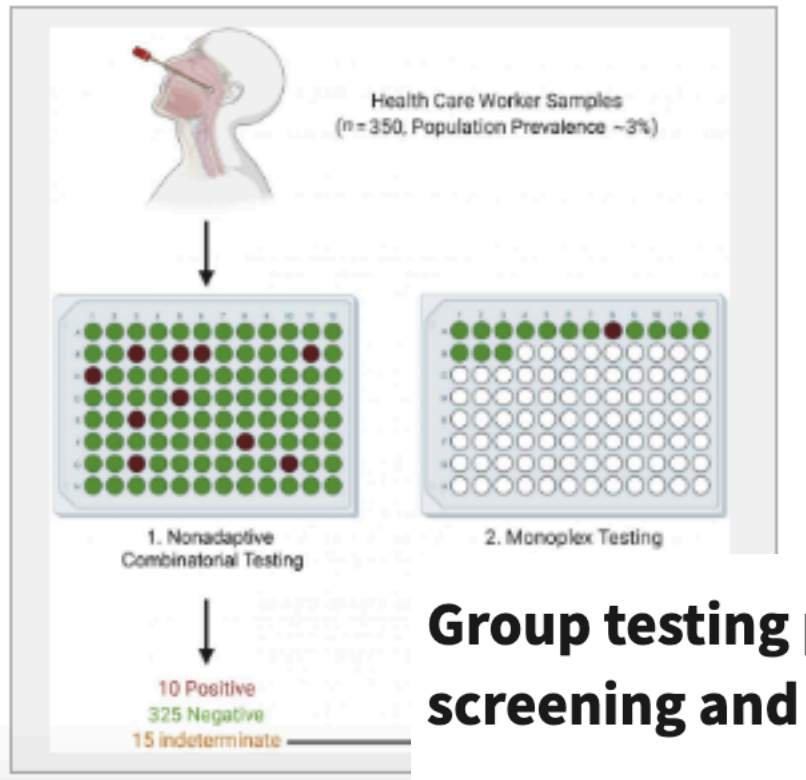
From the Manchester Centre for Genomic Medicine, St. Mary's Hospital, Manchester University NHS Foundation Trust, Manchester, United Kingdom; Division of Evolution and Genomic Sciences,† School of Biological Sciences, University of Manchester, Manchester, United Kingdom; DS Analytics and Machine Learning Ltd.,‡ London, United Kingdom; and Origami Assays,§ Ann Arbor, Michigan*

ting strategy can sp
19 test results for h

Molecular Diagnostics investigators share a
hat maximizes the proportion of samples re

blication

26, 2021 - Fast
VID-19 test results for
s is critical. Investigators
ed a COVID-19 testing
imizes the proportion of
negative results after a single round of
testing, allowing prompt notification of
results. The method also reduces the need
asingly limited test reagents, as
lditional tests are required. Their
is described in *The Journal of*
r Diagnostics, published by Elsevier.



METHODS article

Front. Public Health, 17 August 2021
Sec. Infectious Diseases: Epidemiology and Prevention
Volume 9 - 2021 | https://doi.org/10.3389/fpubh.2021.583377

Group Testing for SARS-CoV-2 Allows for Up to 10-Fold Efficiency Increase Across Realistic Scenarios and Testing Strategies

Updated

Claudio M. Verdun[1,2†] Tim Fuchs[1†] Pavol Harar[3,4†]
Dennis Elbrächter[5†] David S. Fischer[6] Julius Berner[5†]
Philipp Grohs[3,5,7] Fabian J. Theis[1,6] Felix Krahmer[1,8*]

Group testing performance evaluation for SARS-CoV-2 massive scale screening and testing

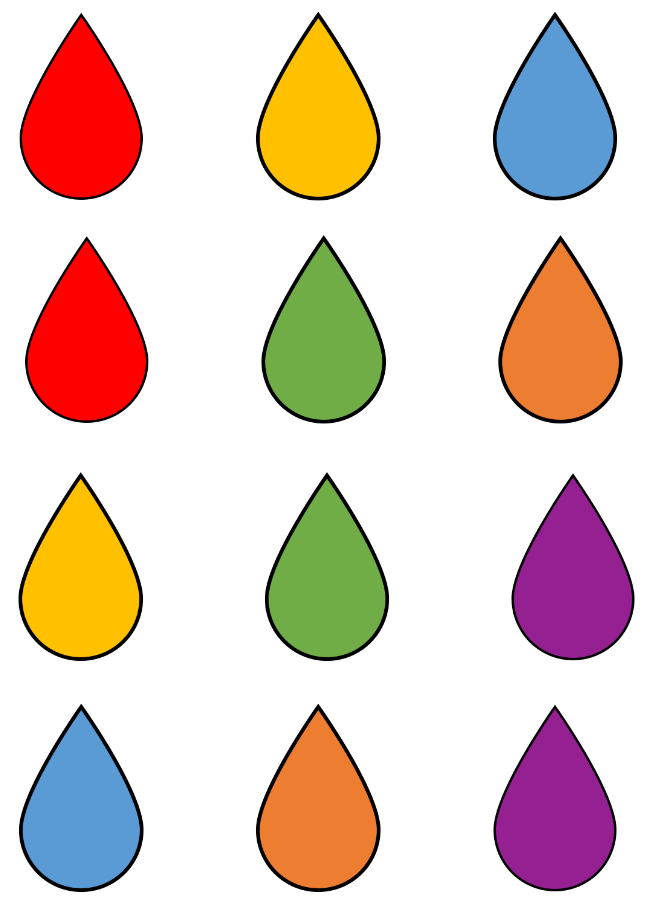
Ozkan Ufuk Nalbantoglu[1,2,✉]

Author information   Article notes   Copyright and License information

PMCID: PMC7330001 PMID: 32615934

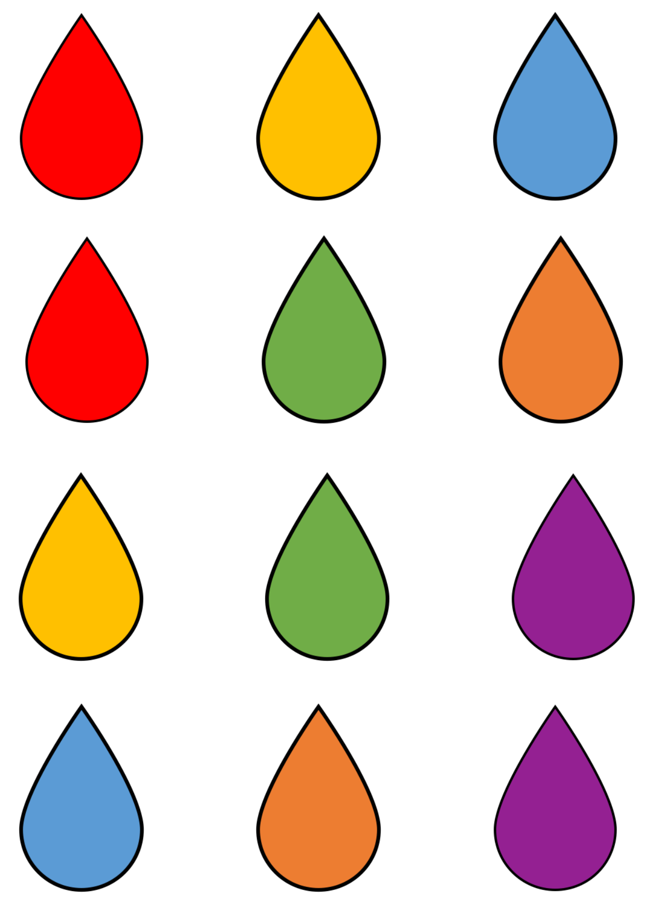
Cover-Free Families

	1	2	3	4	5	6
Test 1	1	1	1	0	0	0
Test 2	1	0	0	1	1	0
Test 3	0	1	0	1	0	1
Test 4	0	0	1	0	1	1



Cover-Free Families

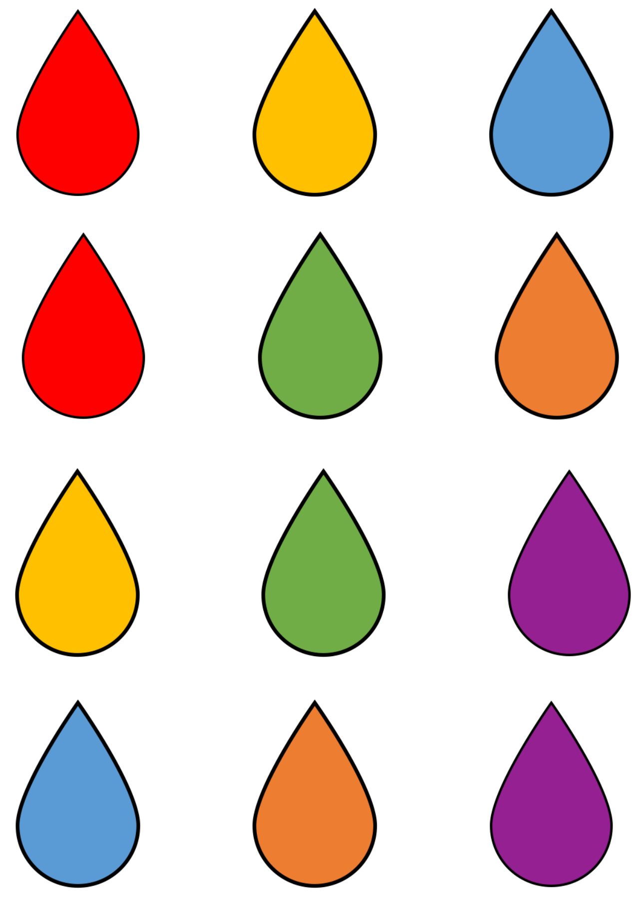
	1	2	3	4	5	6
Test 1	1	1	1	0	0	0
Test 2	1	0	0	1	1	0
Test 3	0	1	0	1	0	1
Test 4	0	0	1	0	1	1



$\boldsymbol{d} - \textbf{CFF(t, n)}$

Cover-Free Families

	1	2	3	4	5	6
Test 1	1	1	1	0	0	0
Test 2	1	0	0	1	1	0
Test 3	0	1	0	1	0	1
Test 4	0	0	1	0	1	1

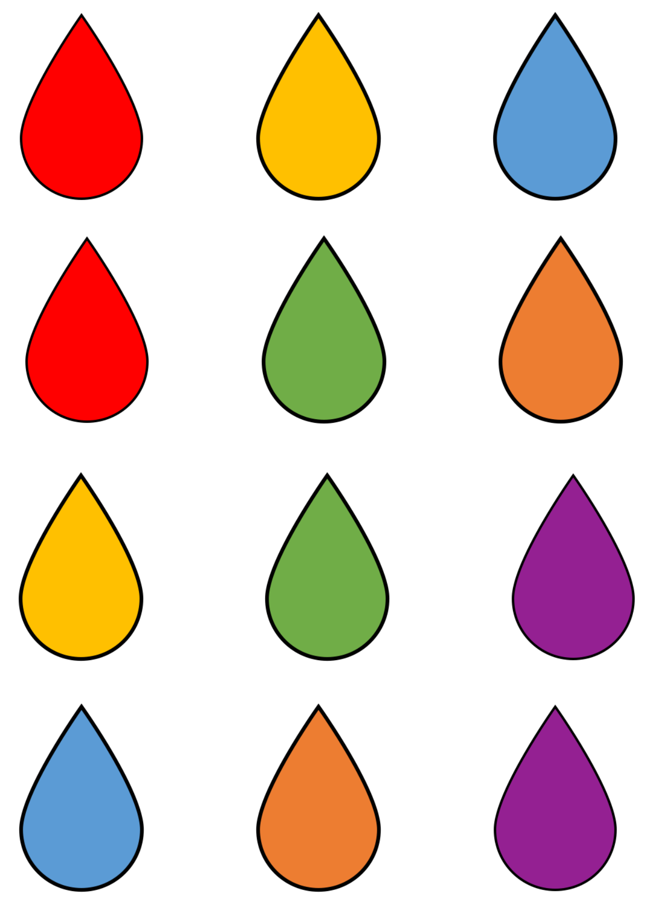


***1* – CFF(4, 6)**

No column is **covered** by any other

Cover-Free Families

	1	2	3	4	5	6
Test 1	1	1	1	0	0	0
Test 2	1	0	0	1	1	0
Test 3	0	1	0	1	0	1
Test 4	0	0	1	0	1	1



FAIL

FAIL

PASS

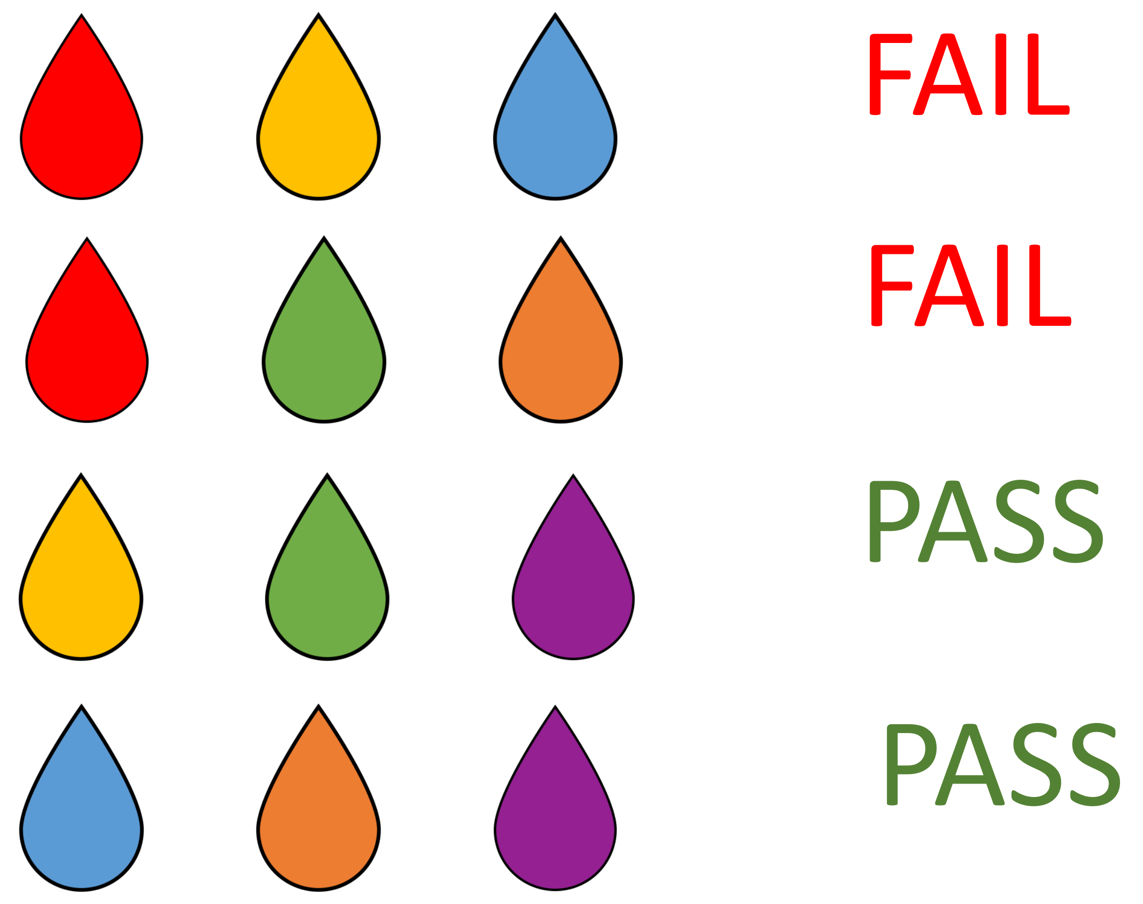
PASS

***1* – CFF(4, 6)**

No column is **covered** by any other

Cover-Free Families

	1	2	3	4	5	6
Test 1	1	1	1	0	0	0
Test 2	1	0	0	1	1	0
Test 3	0	1	0	1	0	1
Test 4	0	0	1	0	1	1

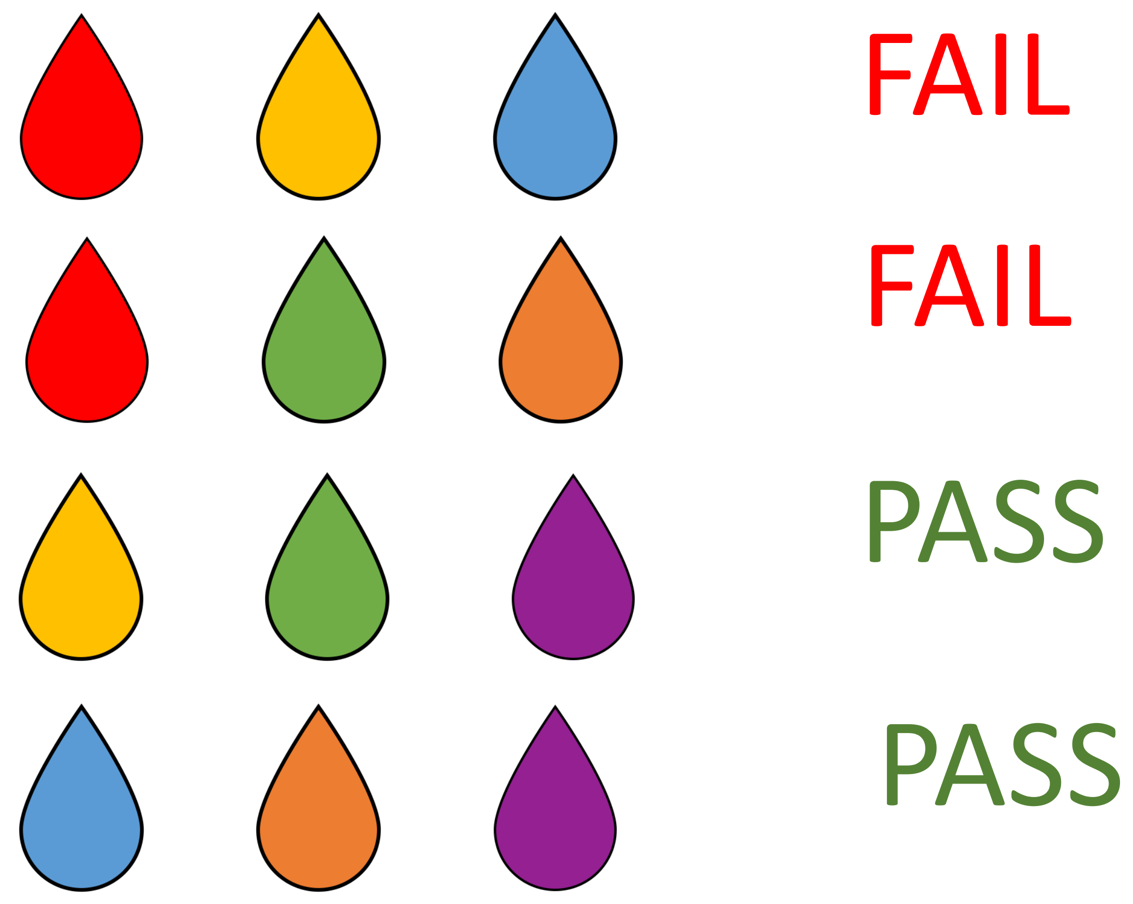


***1* – CFF(4, 6)**

No column is **covered** by any other

Cover-Free Families

	1	2	3	4	5	6
Test 1	1	1	1	0	0	0
Test 2	1	0	0	1	1	0
Test 3	0	1	0	1	0	1
Test 4	0	0	1	0	1	1

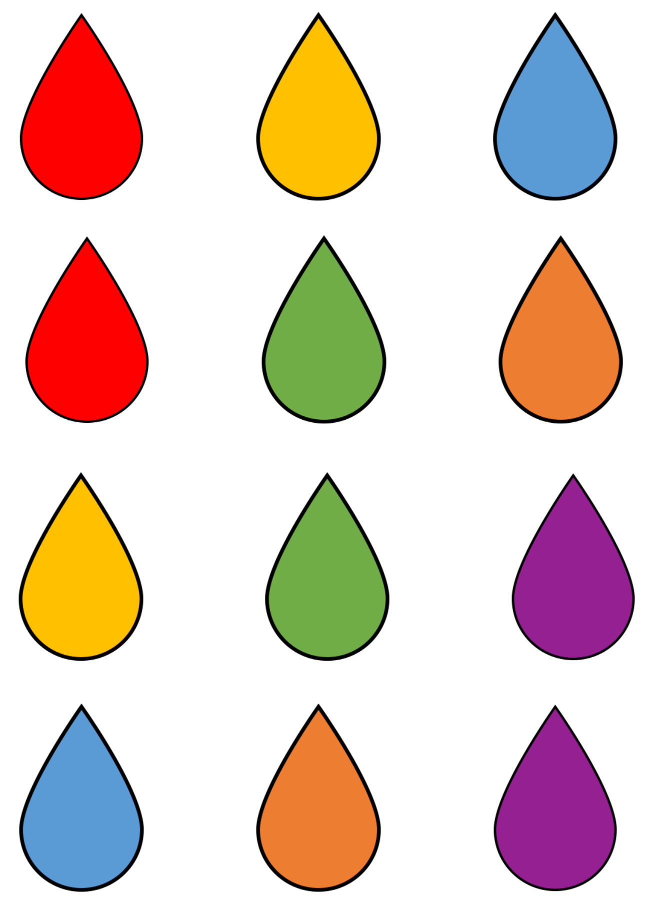


***1* – CFF(4, 6)**

No column is **covered** by any other

Cover-Free Families

	1	2	3	4	5	6
Test 1	1	1	1	0	0	0
Test 2	1	0	0	1	1	0
Test 3	0	1	0	1	0	1
Test 4	0	0	1	0	1	1



FAIL

FAIL

PASS

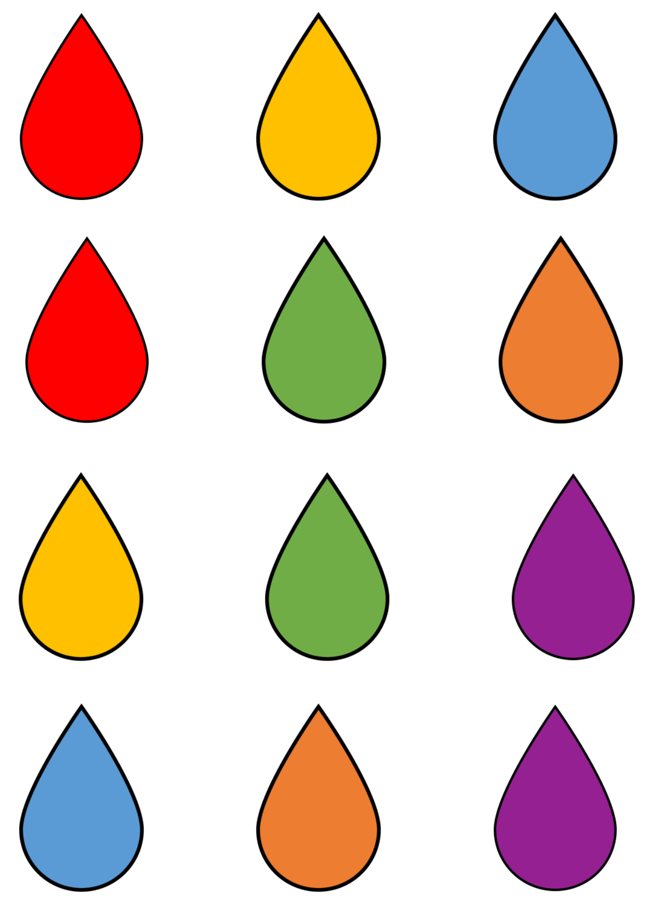
PASS

***1* – CFF(4, 6)**

No column is **covered** by any other

Cover-Free Families

	1	2	3	4	5	6
Test 1	1	1	1	0	0	0
Test 2	1	0	0	1	1	0
Test 3	0	1	0	1	0	1
Test 4	0	0	1	0	1	1



FAIL

FAIL

PASS

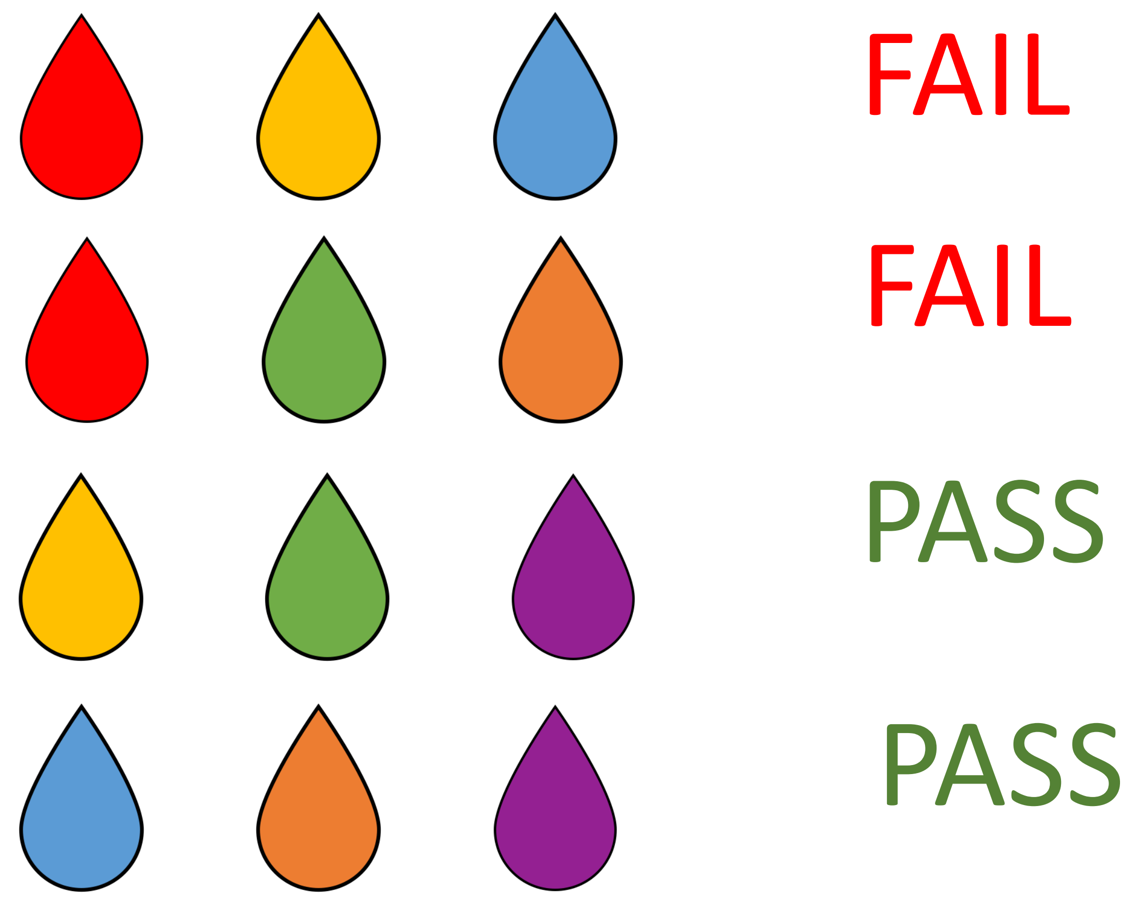
PASS

1 **– CFF(4, 6)**

No column is **covered** by any other

Cover-Free Families

	1	2	3	4	5	6
Test 1	1	1	1	0	0	0
Test 2	1	0	0	1	1	0
Test 3	0	1	0	1	0	1
Test 4	0	0	1	0	1	1

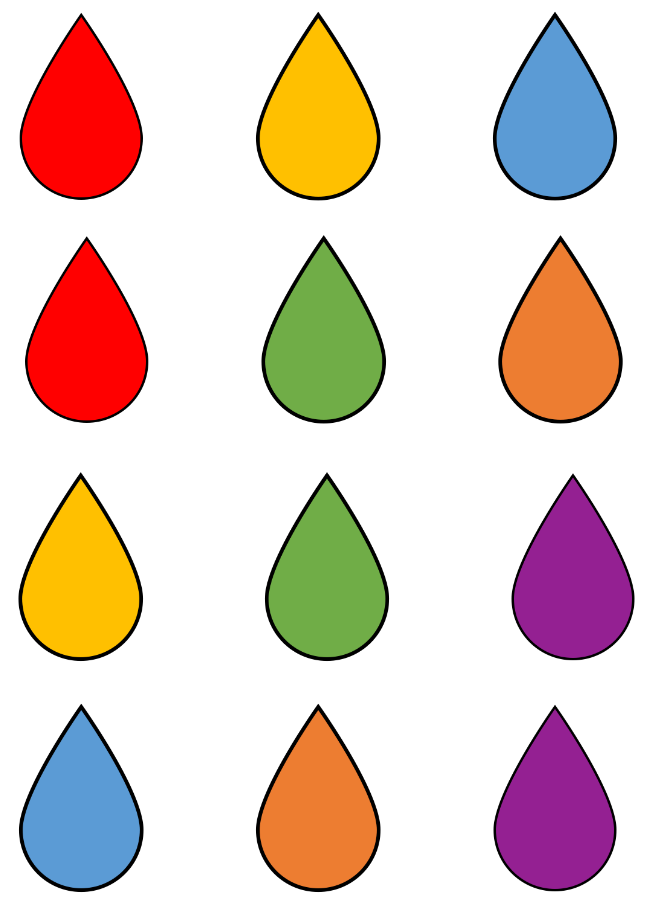


1 – CFF(4, 6)

No column is **covered** by any other

Cover-Free Families

	1	2	3	4	5	6
Test 1	1	1	1	0	0	0
Test 2	1	0	0	1	1	0
Test 3	0	1	0	1	0	1
Test 4	0	0	1	0	1	1



FAIL

PASS

FAIL

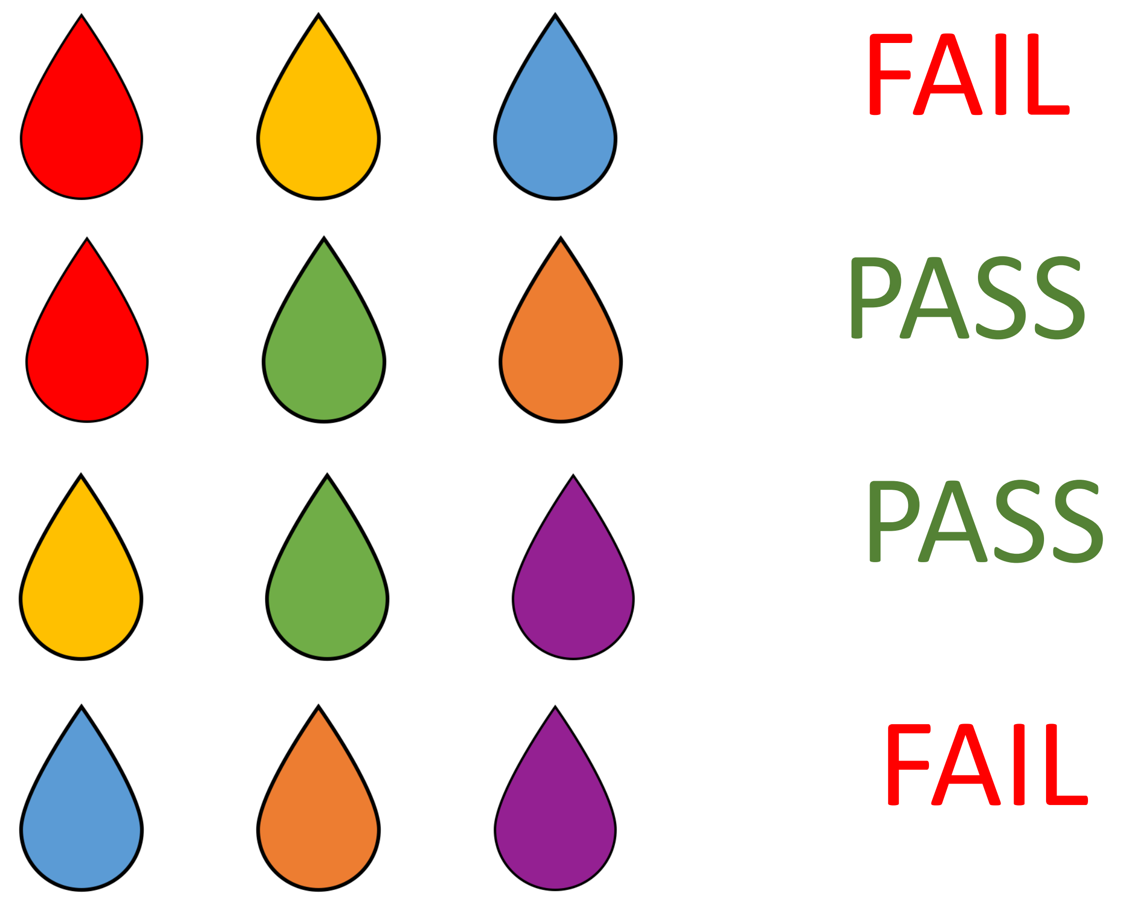
PASS

1 **– CFF(4, 6)**

No column is **covered** by any other

Cover-Free Families

	1	2	3	4	5	6
Test 1	1	1	1	0	0	0
Test 2	1	0	0	1	1	0
Test 3	0	1	0	1	0	1
Test 4	0	0	1	0	1	1



***1* – CFF(4, 6)**

No column is **covered** by any other

Cover-Free Families

	1	2	3	4	5	6
Test 1	1	1	1	0	0	0
Test 2	1	0	0	1	1	0
Test 3	0	1	0	1	0	1
Test 4	0	0	1	0	1	1

1 – CFF(4, 6)

No column is **covered** by any other

Cover-Free Families

	1	2	3	4	5	6
Test 1	1	1	1	0	0	0
Test 2	1	0	0	1	1	0
Test 3	0	1	0	1	0	1
Test 4	0	0	1	0	1	1

1 – CFF(4, 6)

No column is **covered** by any other

Cover-Free Families

	1	2	3	4	5	6
Test 1	1	1	1	0	0	0
Test 2	1	0	0	1	1	0
Test 3	0	1	0	1	0	1
Test 4	0	0	1	0	1	1

1 – CFF(4, 6)

No column is **covered** by any other

Cover-Free Families

	1	2	3	4	5	6
Test 1	1	1	1	0	0	0
Test 2	1	0	0	1	1	0
Test 3	0	1	0	1	0	1
Test 4	0	0	1	0	1	1

***1* – CFF(4, 6)**

No column is **covered** by any other

Cover-Free Families

	1	**2**	**3**	**4**	**5**	**6**	
Test 1	1	1	1	0	0	0	FAIL
Test 2	1	0	0	1	1	0	FAIL
Test 3	0	1	0	1	0	1	FAIL
Test 4	0	0	1	0	1	1	FAIL

1 – CFF(4, 6)

This is not a 2-CFF!

Cover-Free Families

	1	2	3	4	5	6	7	8	9	10	11	12
Test 1	1			1			1			1		
Test 2	1				1			1			1	
Test 3	1					1			1			1
Test 4		1		1					1		1	
Test 5		1			1		1					1
Test 6		1				1		1		1		
Test 7			1	1				1				1
Test 8			1		1				1	1		
Test 9			1			1	1				1	

2 - CFF(9, 12)

No column is **covered** by the union of any **two** others

Cover-Free Families

	1	2	3	4	5	6	7	8	9	10	11	12
Test 1	1			1			1			1		
Test 2	1				1			1			1	
Test 3	1					1			1			1
Test 4		1		1					1		1	
Test 5		1			1		1					1
Test 6		1				1		1		1		
Test 7			1	1				1				1
Test 8			1		1				1	1		
Test 9			1			1	1				1	

2 - CFF(9, 12)

No column is **covered** by the union of any **two** others

Cover-Free Families

	1	2	3	4	5	6	7	8	9	10	11	12
Test 1	1			1			1			1		
Test 2	1				1			1			1	
Test 3	1					1			1			1
Test 4		1		1					1		1	
Test 5		1			1		1					1
Test 6		1				1		1		1		
Test 7			1	1				1				1
Test 8			1		1				1	1		
Test 9			1			1	1				1	

2 - CFF(9, 12)

No column is **covered** by the union of any **two** others

Cover-Free Families

	1	2	3	4	5	6	7	8	9	10	11	12
Test 1	1			1			1			1		
Test 2	1				1			1			1	
Test 3	1					1			1			1
Test 4		1		1					1		1	
Test 5		1			1		1					1
Test 6		1				1		1		1		
Test 7			1	1				1				1
Test 8			1		1				1	1		
Test 9			1			1	1				1	

2 - CFF(9, 12)

No column is **covered** by the union of any **two** others

Cover-Free Families

X	B_1	B_2	B_3	B_4	B_5	B_6
1	1	1	1	0	0	0
2	1	0	0	1	1	0
3	0	1	0	1	0	1
4	0	0	1	0	1	1

Definition: Let d be a positive integer. A *d*-cover-free family, denoted $d - CFF(t, n)$, is a set system $\mathscr{F} = (X, \mathscr{B})$ with $|X| = t$ and $\mathscr{B} = \{B_1, B_2, \ldots, B_n\}$ such that for any $d + 1$ subsets $B_{i_0}, B_{i_1}, \ldots, B_{i_d} \in \mathscr{B}$, we have:

$$\left| B_{i_0} \setminus \left(\bigcup_{j=1}^{d} B_{i_j} \right) \right| \geq 1.$$

No element is ***covered*** by the union of any *d* others.

* Equivalent to disjunct matrices and superimposed codes.

Cover-Free Families

$X = \{1,2,\ldots,9\}$

$\mathscr{B} = \{B_1, \ldots, B_{12}\}$

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}	B_{11}	B_{12}
1	1			1			1			1		
2	1				1			1			1	
3	1					1			1			1
4		1		1					1		1	
5		1			1		1					1
6		1				1		1		1		
7			1	1				1				1
8			1		1				1	1		
9			1			1	1				1	

$B_1 \cup B_2 = \{1,2,3,4,5,6\}$

$B_3 - (B_1 \cup B_2) = \{7,8,9\}$

$B_4 - (B_1 \cup B_2) = \{7\}$

$B_5 - (B_1 \cup B_2) = \{8\}$

$B_6 - (B_1 \cup B_2) = \{9\}$

$B_7 - (B_1 \cup B_2) = \{9\}$

$B_8 - (B_1 \cup B_2) = \{7\}$

$B_9 - (B_1 \cup B_2) = \{8\}$

$B_{10} - (B_1 \cup B_2) = \{8\}$

$B_{11} - (B_1 \cup B_2) = \{9\}$

$B_{12} - (B_1 \cup B_2) = \{7\}$

Constructing CFFs

Bounds

n

t

1	1	1	0	0	0
1	0	0	1	1	0
0	1	0	1	0	1
0	0	1	0	1	1

- **Minimize** *t* for given *n* and *d*
  - $t(d, n) = \min\{t : \exists\ d\text{-CFF}(t, n)\}$
- When $d = 1$, Sperner's construction gives $t(1,n) \sim \log_2 n$ when $n \to \infty$;
- For $d \geq 2$, the best known **lower bound** on t for *d*-CFF(*t*, *n*) is:

$$t(d, n) \geq c\frac{d^2}{\log d}\log n$$

Sperner construction

$d = 1$

- For a given *n*, choose the smallest *t* such that

$$n \leq \binom{t}{\lfloor t/2 \rfloor}$$

- Consider $X = \{1,2,\ldots,t\}$
- $\mathscr{B} = \{B_1, B_2, \ldots, B_n\}$ is the list of all subsets of X with cardinality $\lfloor t/2 \rfloor$

Sperner construction

$d = 1$

For n = 6, t = 4:

$$6 = \binom{4}{2}$$

For n = 100, t = 9:

$$100 \leq \binom{9}{4}$$

For n = 1500, t = 13:

$$1500 \leq \binom{13}{6}$$

$X = \{1,2,3,4\}$

$B_1 = \{1,2\}$

$B_2 = \{1,3\}$

$B_3 = \{1,4\}$

$B_4 = \{2,3\}$

$B_5 = \{2,4\}$

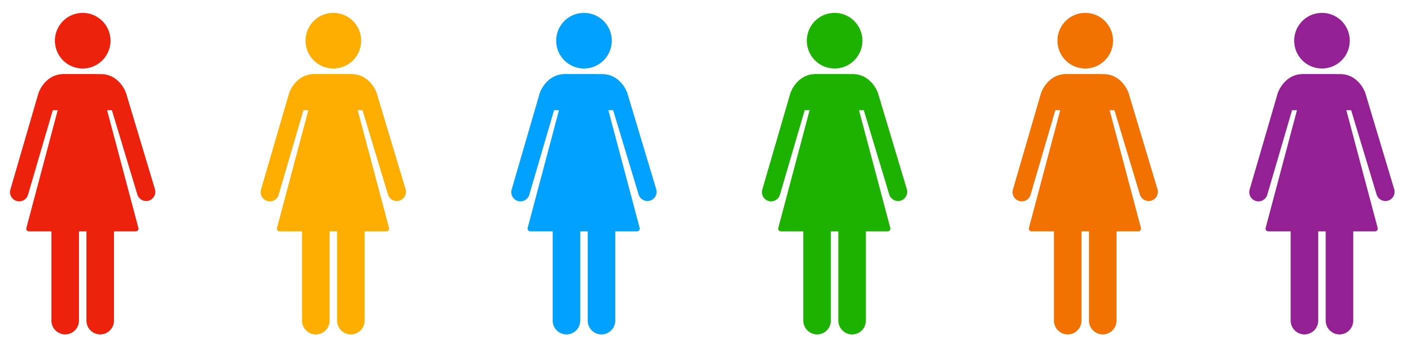
$B_6 = \{3,4\}$

	B_1	B_2	B_3	B_4	B_5	B_6
1	1	1	1	0	0	0
2	1	0	0	1	1	0
3	0	1	0	1	0	1
4	0	0	1	0	1	1

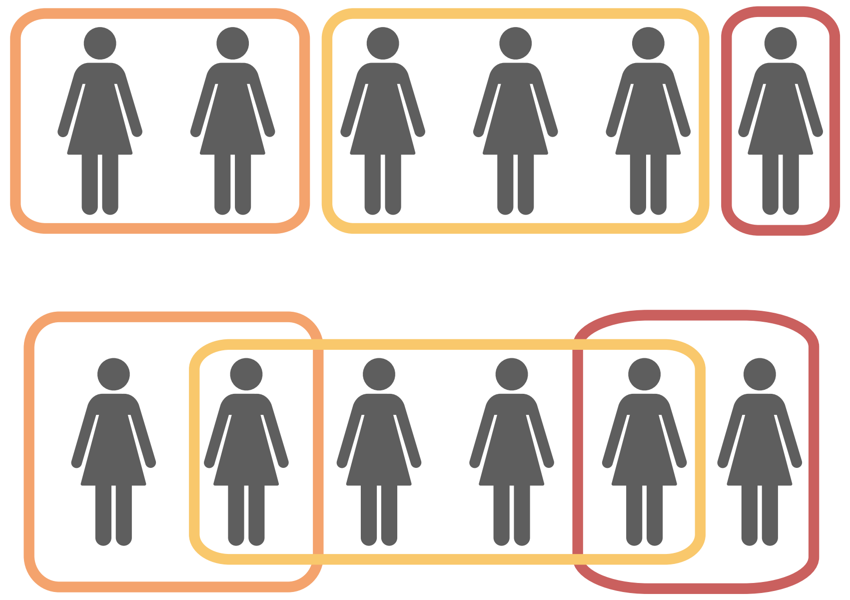
***1* – CFF(4, 6)**

In this talk

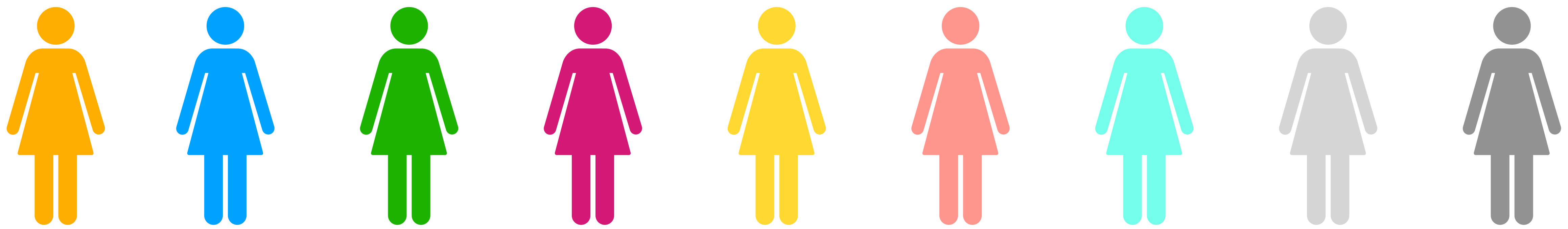
1	1	1	0	0	0
1	0	0	1	1	0
0	1	0	1	0	1
0	0	1	0	1	1



- The problem of **combinatorial group testing** in pandemic screening
- Study of ***CFFs on hypergraphs***
  - Non-overlapping edges
  - Overlapping edges
  - Some other cases
- Applications in cryptography

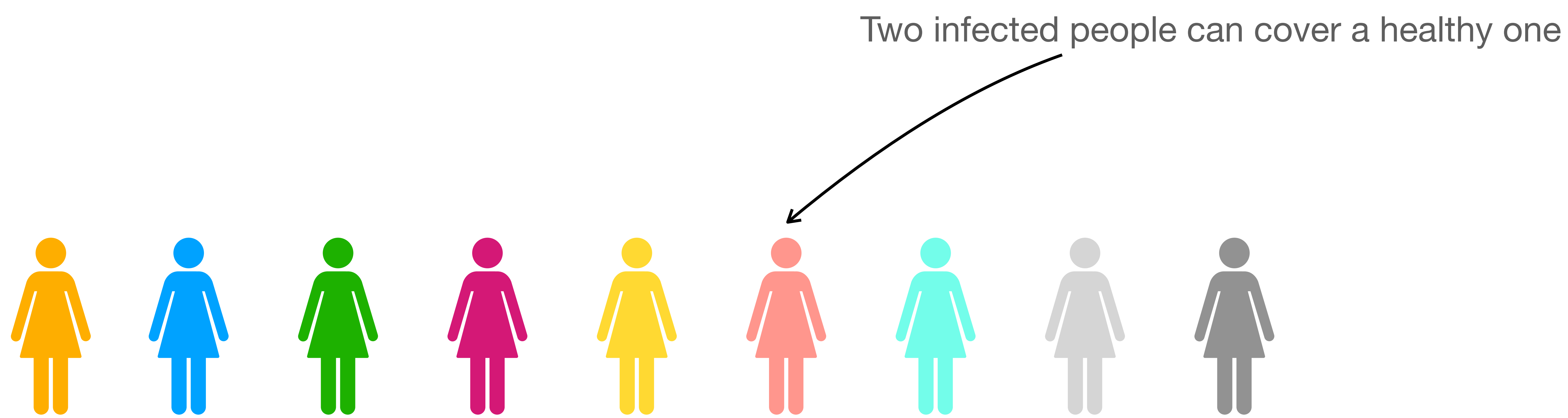


1. Bardini Idalino, T., Moura, L. Structure-Aware Combinatorial Group Testing: A New Method for PandemicScreening. In: IWOCA 2022. Lecture Notes in Computer Science, vol 13270. Springer, Cham. https://doi.org/10.1007/978-3-031-06678-8
2. Bardini Idalino, T., Moura, L. Combinatorial group testing and cover-free families on hypergraphs. 2025[Unpublished manuscript].



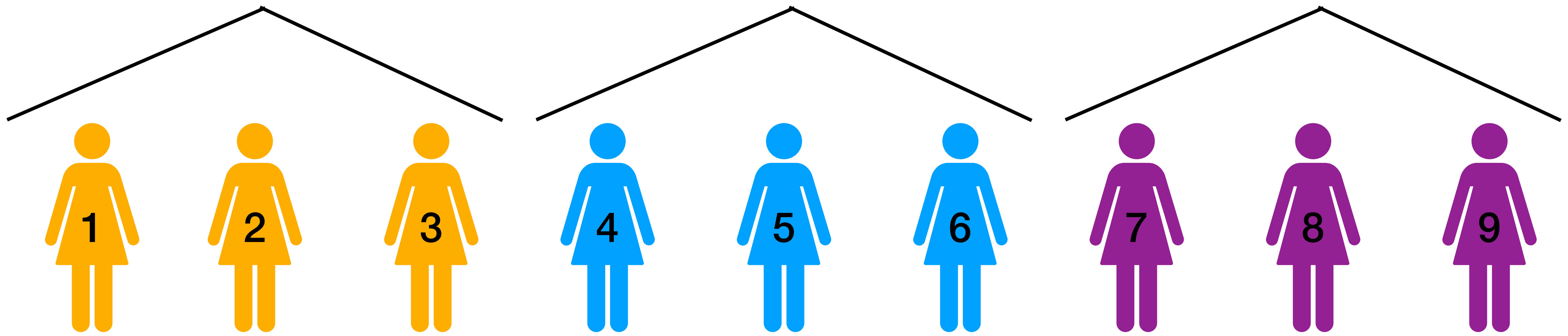
1	1	1	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	1	1	1
1	0	0	0	0	1	0	1	0
0	1	0	1	0	0	0	0	1
0	0	1	0	1	0	1	0	0

***1* – CFF(6, 9)**



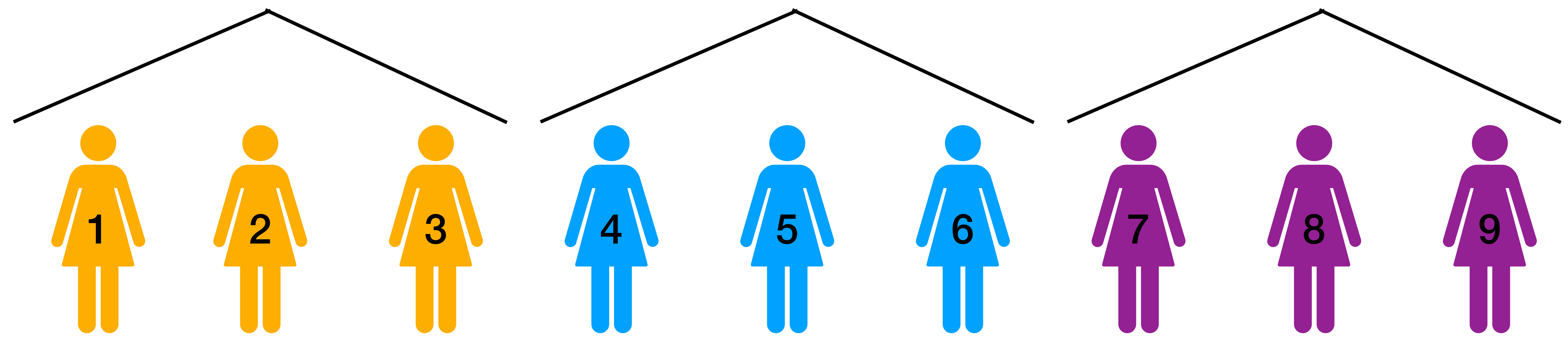
1	1	1	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	1	1	1
1	0	0	0	0	1	0	1	0
0	1	0	1	0	0	0	0	1
0	0	1	0	1	0	1	0	0

1 – CFF(6, 9)



1	1	1	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	1	1	1
1	0	0	0	0	1	0	1	0
0	1	0	1	0	0	0	0	1
0	0	1	0	1	0	1	0	0

***1* – CFF(6, 9)**

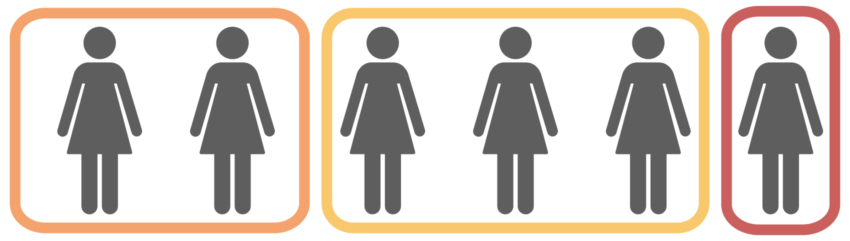


1	1	1	0	0	0	0	0	0	FAIL
0	0	0	1	1	1	0	0	0	PASS
0	0	0	0	0	0	1	1	1	PASS
1	0	0	0	0	1	0	1	0	FAIL
0	1	0	1	0	0	0	0	1	FAIL
0	0	1	0	1	0	1	0	0	PASS

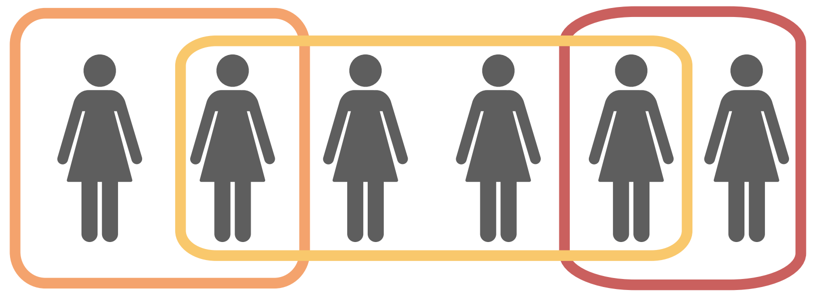
1 – CFF(6, 9)

CFFs on hypergraphs

Model the problem using **hypergraphs** $\mathscr{H} = (V, \mathscr{E})$



- Vertices $\rightarrow$ items/columns of the CFF
- Edges $\rightarrow$ potential clusters of items / communities

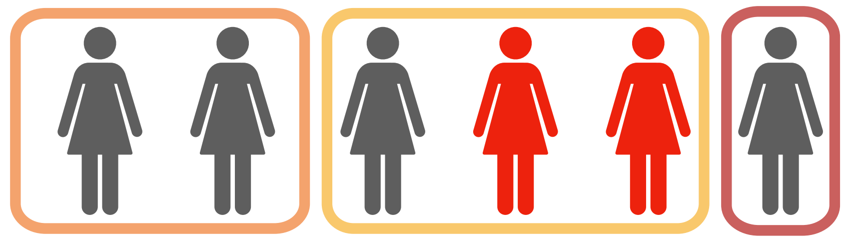


Propose constructions that take $\mathscr{H}$ into consideration

- $(\mathscr{E}, r) - CFF(t, n)$ (vertex-identifying CFF)
- $(\mathscr{E}, r) - ECFF(t, n)$ (edge-identifying CFF)

Vertex-identifying CFFs

$(\mathscr{E}, r) - CFF(t, n)$



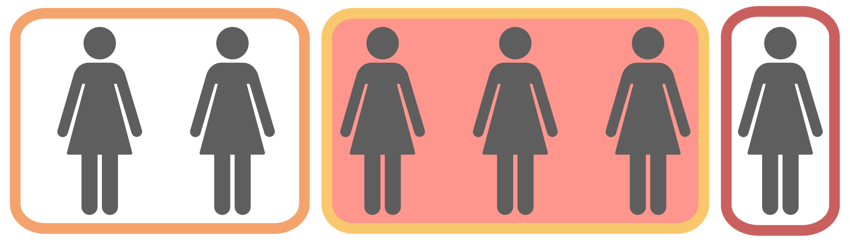
Definition: Let $n, t > 0$ and $r \geq 0$ be integers. Let $\mathscr{H} = ([1,n], \mathscr{E})$ be a hypergraph with *n* vertices and *m* edges. A set system $\mathscr{F} = ([1,t], \mathscr{B})$, $\mathscr{B} = \{B_1, \ldots, B_n\}$ is a $(\mathscr{E}, r) - CFF(t, n)$ if, for any set of *r* edges $\{e_1, \ldots, e_r\} \subseteq \mathscr{E}$, and for any $I \subseteq \cup_{j=1}^{r} e_j$ and any $i_0 \in [1,n] \backslash I$, we have

$$\left| B_{i_0} \backslash \left(\bigcup_{i \in I} B_i \right) \right| \geq 1.$$

In other words: we can identify all infected vertices, as long as there are at most *r* infected edges that jointly contain them

Edge-identifying CFFs

$(\mathcal{E}, r) - ECFF(t, n)$



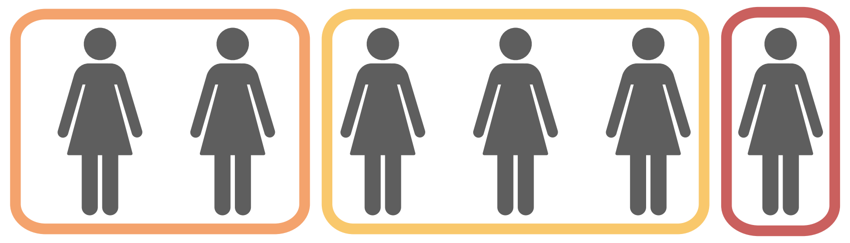
Definition: Let $r, n, t, \mathcal{H}, \mathcal{F}$ be as in the previous definition. The set systems is an $(\mathcal{E}, r) - ECFF(t, n)$ if for any set of ℓ edges $\{e_1, \ldots, e_\ell\} \subseteq \mathcal{E}$, $\ell \leq r$, and any $i_0 \notin E = \cup_{j=1}^{\ell} e_j$, we have

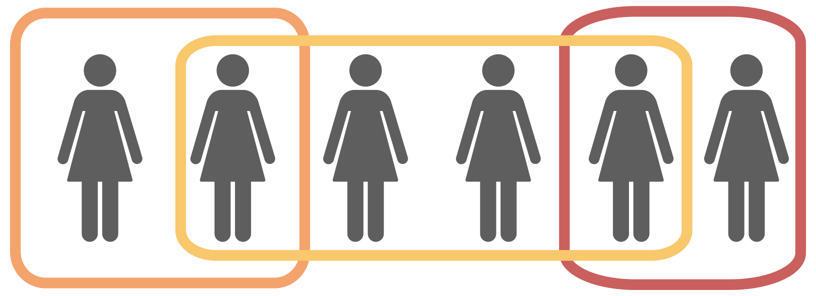
$$\left| B_{i_0} \setminus \left(\bigcup_{i \in E} B_i \right) \right| \geq 1.$$

In other words: we can identify all infected edges, as long as there are at most *r* of them

A bit more detail...

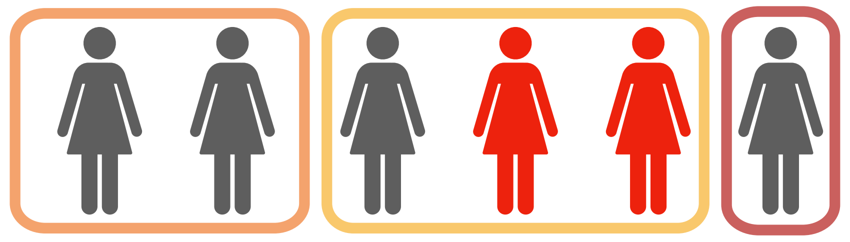
- Overlapping and non-overlapping edges:



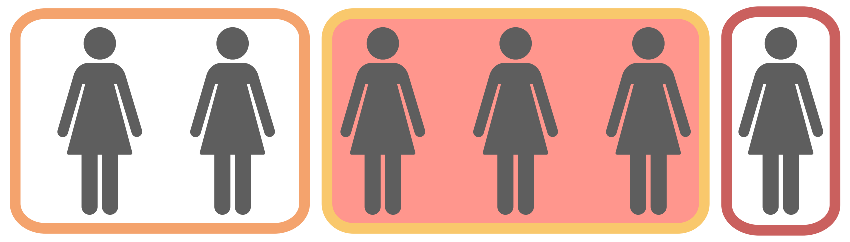


- Minimize the number of rows t:

$$t(r, \mathscr{E}) = \min\{t : \exists\ (\mathscr{E}, r)\text{-CFF}(t, n)\}$$



$$t_E(r, \mathscr{E}) = \min\{t : \exists\ (\mathscr{E}, r)\text{-ECFF}(t, n)\}$$

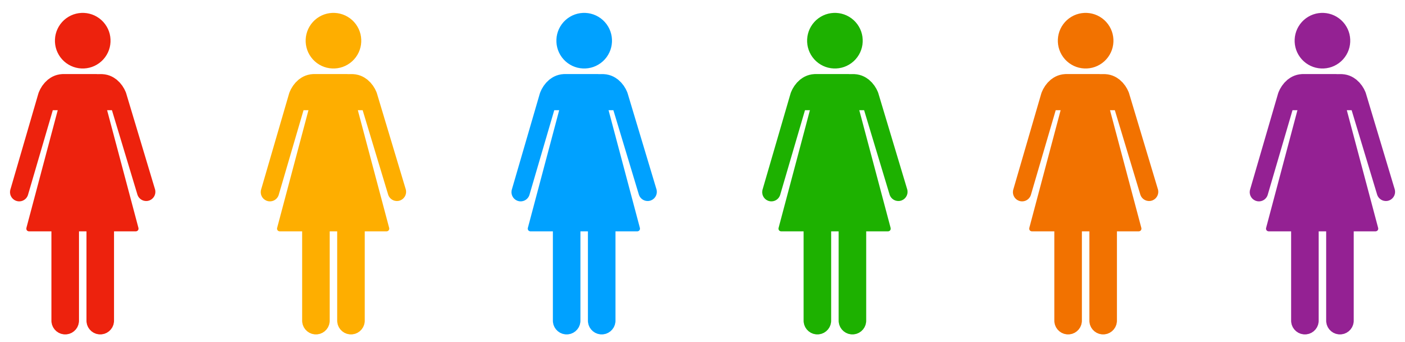


Related Work

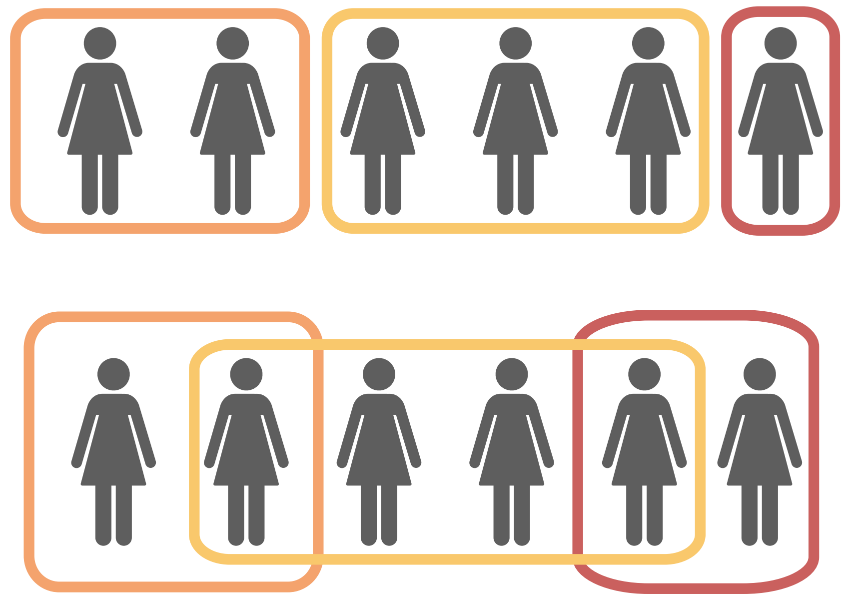
- Several works on CGT for COVID-19 testing
- Some solutions modeled as hypergraphs
  - Community-aware group testing (Nikolopoulos et al., 2021)
    - Connected and overlapping communities
  - Generalized group testing (Gonen et al., 2022 , Vorobyev, 2022, De Bonis, 2024)
    - Edges are all potentially contaminated sets
- **Our work**: focus on the construction of CFFs on hypergraphs:
  - *Variable* CFFs in Cryptography (Idalino, 2019)
  - Structure-Aware Combinatorial Group Testing (Idalino, Moura, 2022)
  - Combinatorial group testing and cover-free families on hypergraphs (Idalino, Moura, 2025?)

In this talk

1	1	1	0	0	0
1	0	0	1	1	0
0	1	0	1	0	1
0	0	1	0	1	1



- The problem of **combinatorial group testing** in pandemic screening
- Study of ***CFFs on hypergraphs***
  - Non-overlapping edges
  - Overlapping edges
  - Some other cases
- Applications in cryptography

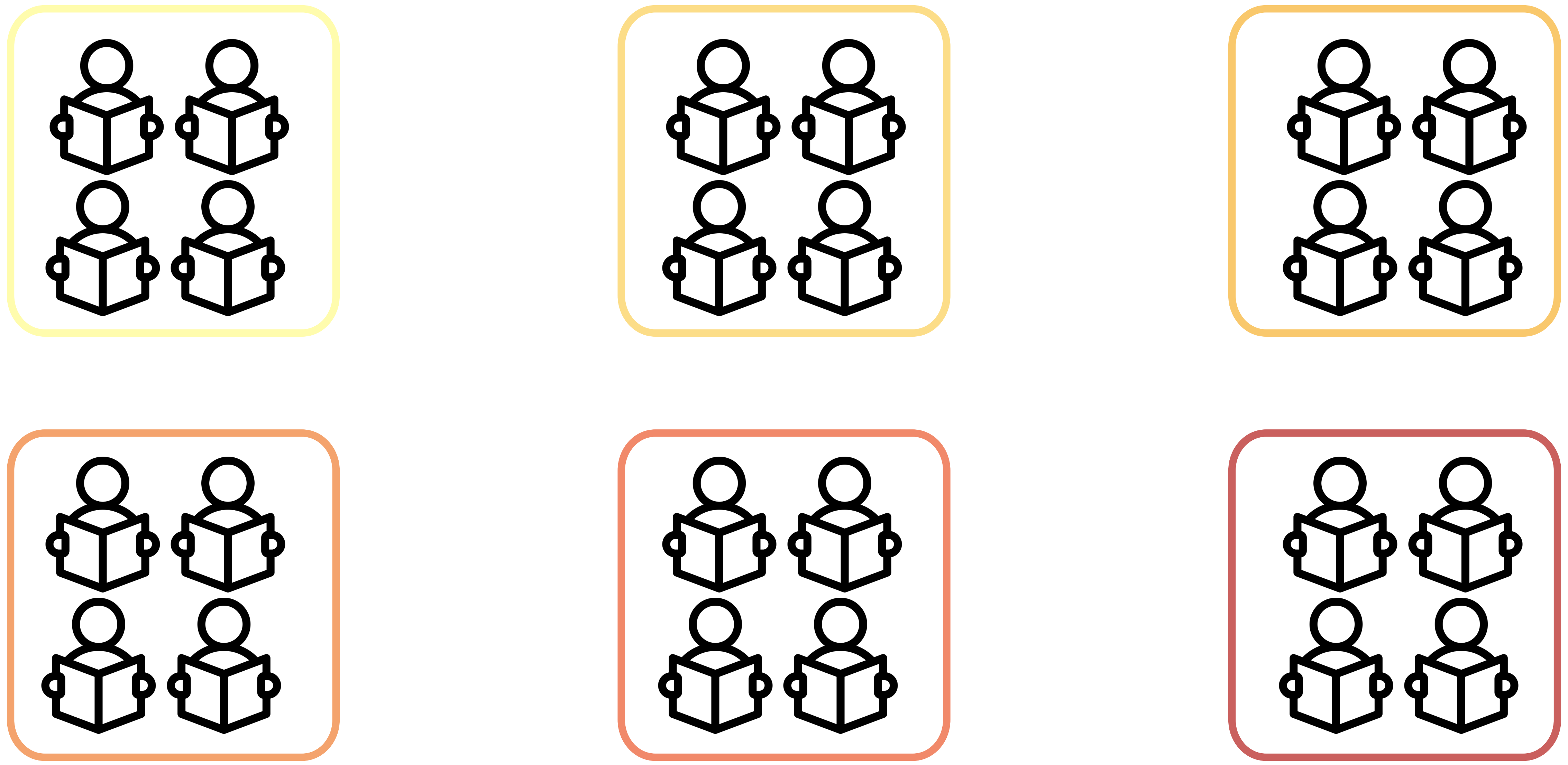


1. Bardini Idalino, T., Moura, L. Structure-Aware Combinatorial Group Testing: A New Method for PandemicScreening. In: IWOCA 2022. Lecture Notes in Computer Science, vol 13270. Springer, Cham. https://doi.org/10.1007/978-3-031-06678-8

2. Bardini Idalino, T., Moura, L. Combinatorial group testing and cover-free families on hypergraphs. 2025 [Unpublished manuscript].

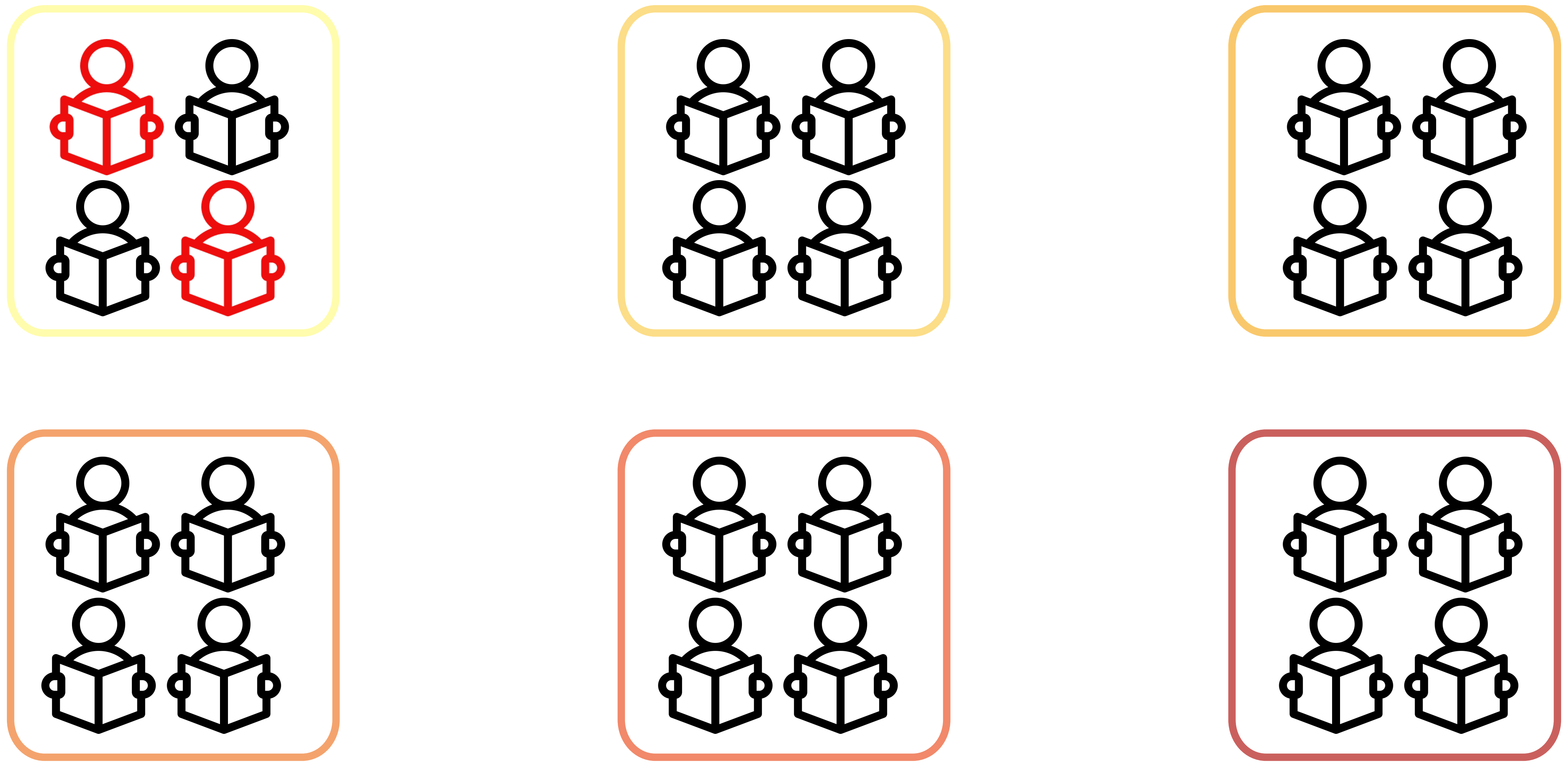
The school problem

Non-overlapping edges



The school problem

Non-overlapping edges, r = 1



The school problem

Non-overlapping edges, r = 1

One column per classroom

1	1	1	0	0	0
1	0	0	1	1	0
0	1	0	1	0	1
0	0	1	0	1	1

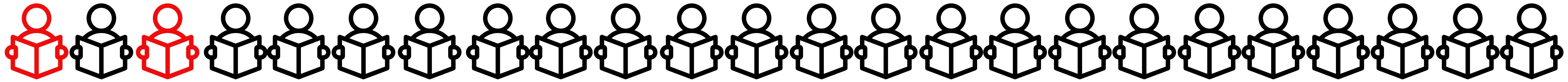
The school problem

Non-overlapping edges, r = 1

Classroom 1				Classroom 2				Classroom 3				Classroom 4				Classroom 5				Classroom 6			
1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1

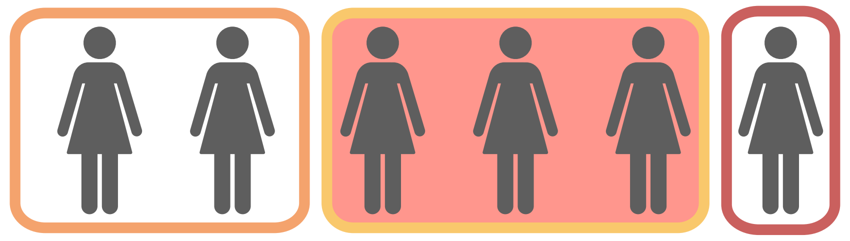
The school problem

Non-overlapping edges, r = 1

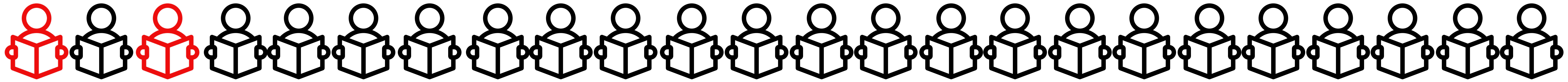


1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1

The school problem



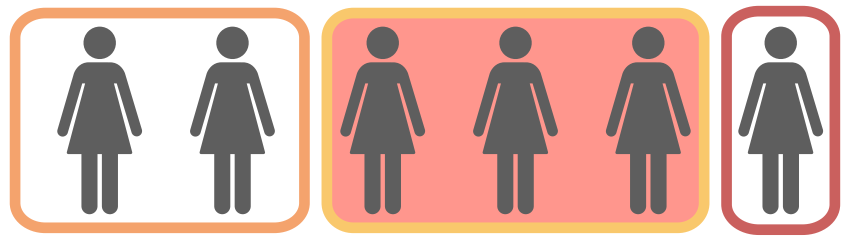
Non-overlapping edges, r = 1



1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	FAIL
1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	FAIL
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	PASS
0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1	PASS

$$(\mathscr{E},1) - ECFF(4,24)$$

Edge-identifying CFF



Generalizing the idea..

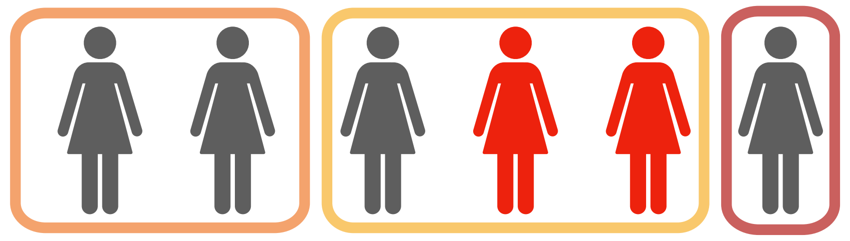
$(\mathscr{E},1) - ECFF(t,n)$

- Let $\mathscr{H} = ([1,n], \mathscr{E})$ be a hypergraph with m disjoint edges of cardinality k
- Construct $(\mathscr{E},1) - ECFF(t,n)$ as follows:
  - **Step 1**: pick a good $1 - CFF(t,m)$
  - **Step 2**: repeat each column k times
- We have $t_E(1,\mathscr{E}) = t(1,m) = \Theta(\log m)$

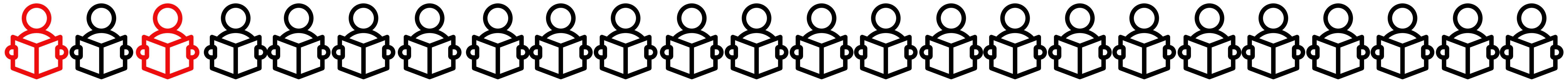
e_1	e_2	e_3	e_4	e_5	e_6
1	1	1	0	0	0
1	0	0	1	1	0
0	1	0	1	0	1
0	0	1	0	1	1

1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1

The school problem



Non-overlapping edges, r = 1

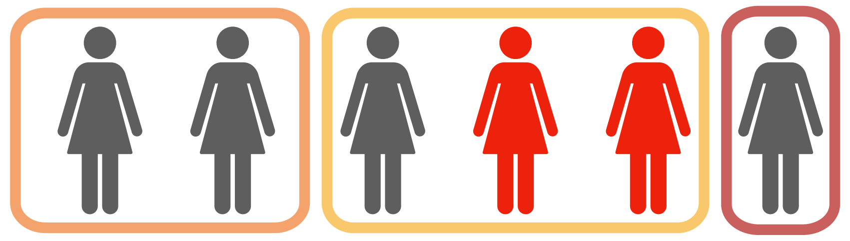


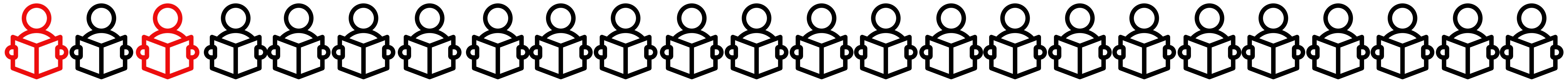
1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1

1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0
0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0
0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1

The school problem

Non-overlapping edges, r = 1





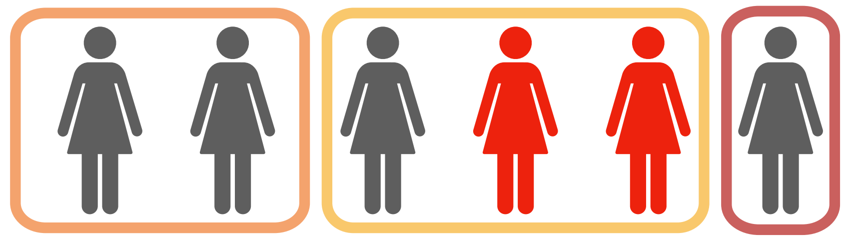
1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	FAIL
1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	FAIL
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	PASS
0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1	PASS

1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	FAIL
0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	PASS
0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	FAIL
0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	PASS

$$(\mathscr{E},1) - CFF(8,24)$$

Vertex-identifying CFF

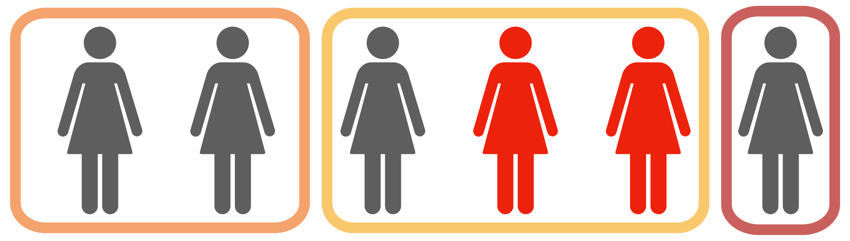
Generalizing the idea..



$(\mathscr{E},1) - CFF(t,n)$

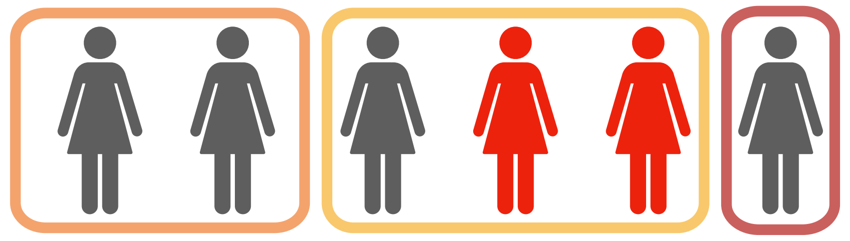
- Let $\mathscr{H} = ([1,n], \mathscr{E})$ be a hypergraph with *m* disjoint edges of cardinality *k*
- Construct $(\mathscr{E},1) - CFF(t,n)$ as follows:
  - **Step 1**: pick a $1 - CFF(t,m)$
  - **Step 2**: repeat each column *k* times
  - **Step 3**: append *m* identity matrices of dimension *k*
- We have $t(1,\mathscr{E}) \leq t(1,m) + k = \Theta(\log m + k)$

Wait.. that is actually good!



$(\mathscr{E},1) - CFF(t,n)$

- Let $\mathscr{H} = ([1,n], \mathscr{E})$ be a hypergraph with *m* disjoint edges of cardinality *k*
- We constructed an $(\mathscr{E},1) - CFF(t,n)$ with only $\Theta(\log m + k)$ tests
  - to test $n = mk$ vertices
  - and identify up to *k* infected ones, as long as they are all inside $r = 1$ edge
- We would need a traditional *k*-CFF(t, n) to identify *k* infections
  - $t(d,n) \geq c\dfrac{k^2}{\log k}\log n$



Wait.. that is actually good!

Comparison with traditional $k - CFF(t, n)$

Total number of students

Number of classrooms

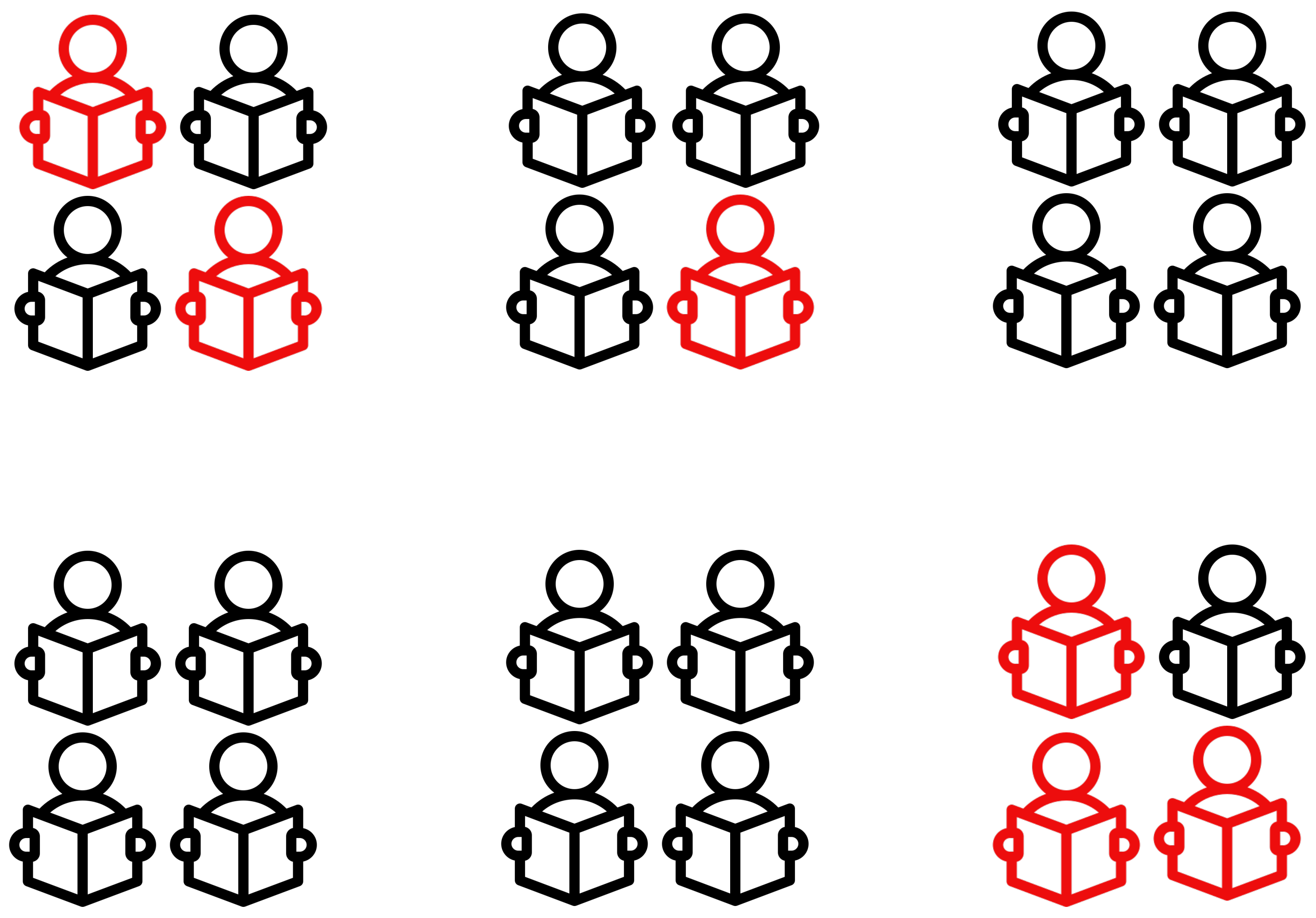
Classroom size

n	m	k	$(\mathscr{E},1) - CFF(t,n)$	$k - CFF(t,n)$
100	10	10	15	66
200	10	20	25	180
300	**10**	**30**	**35**	**231**
100	20	5	11	21
200	20	10	16	66
400	20	20	26	231

Lower bound

The school problem

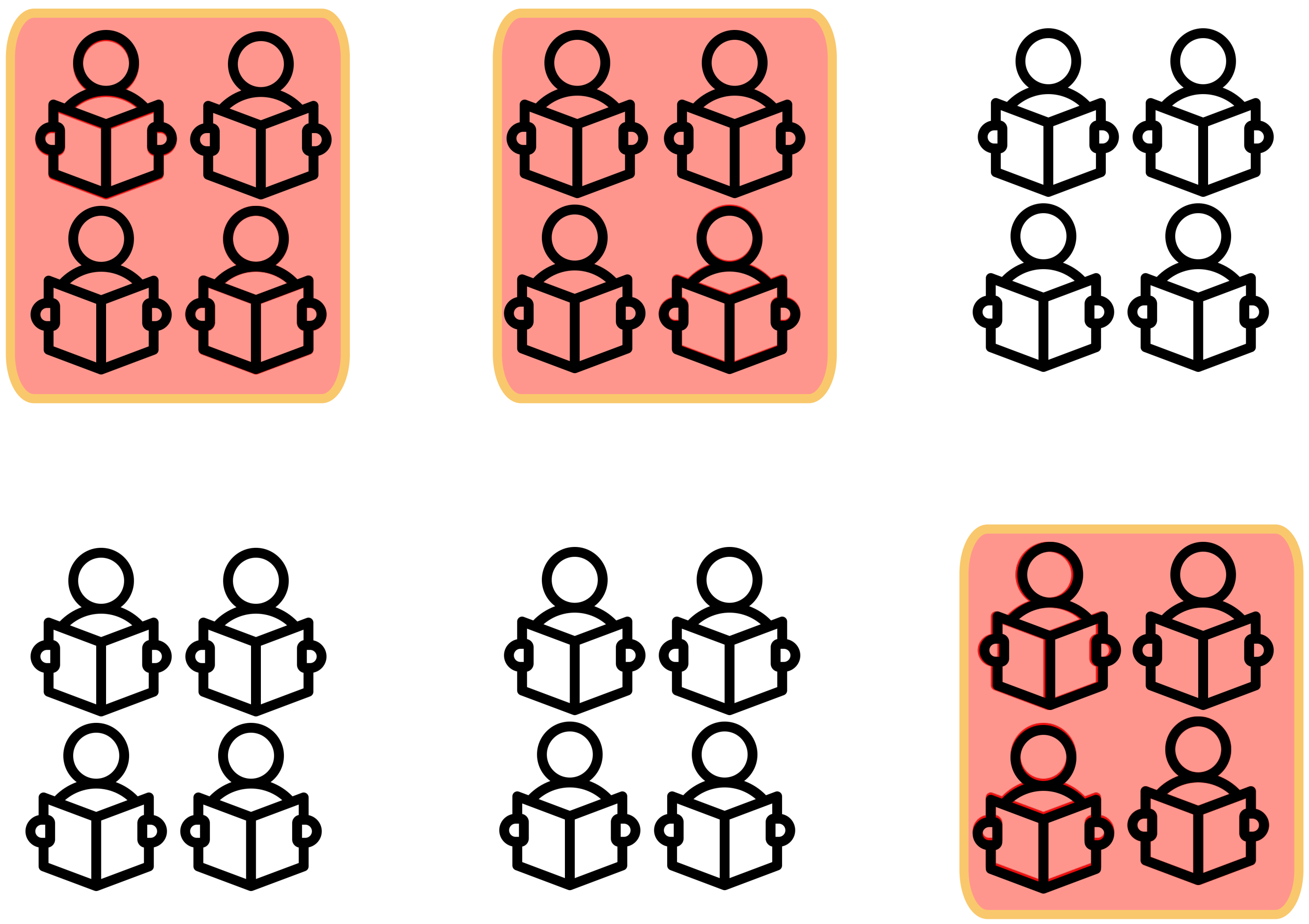
What if more classrooms are infected?



- This is the case $r \geq 2$

The school problem

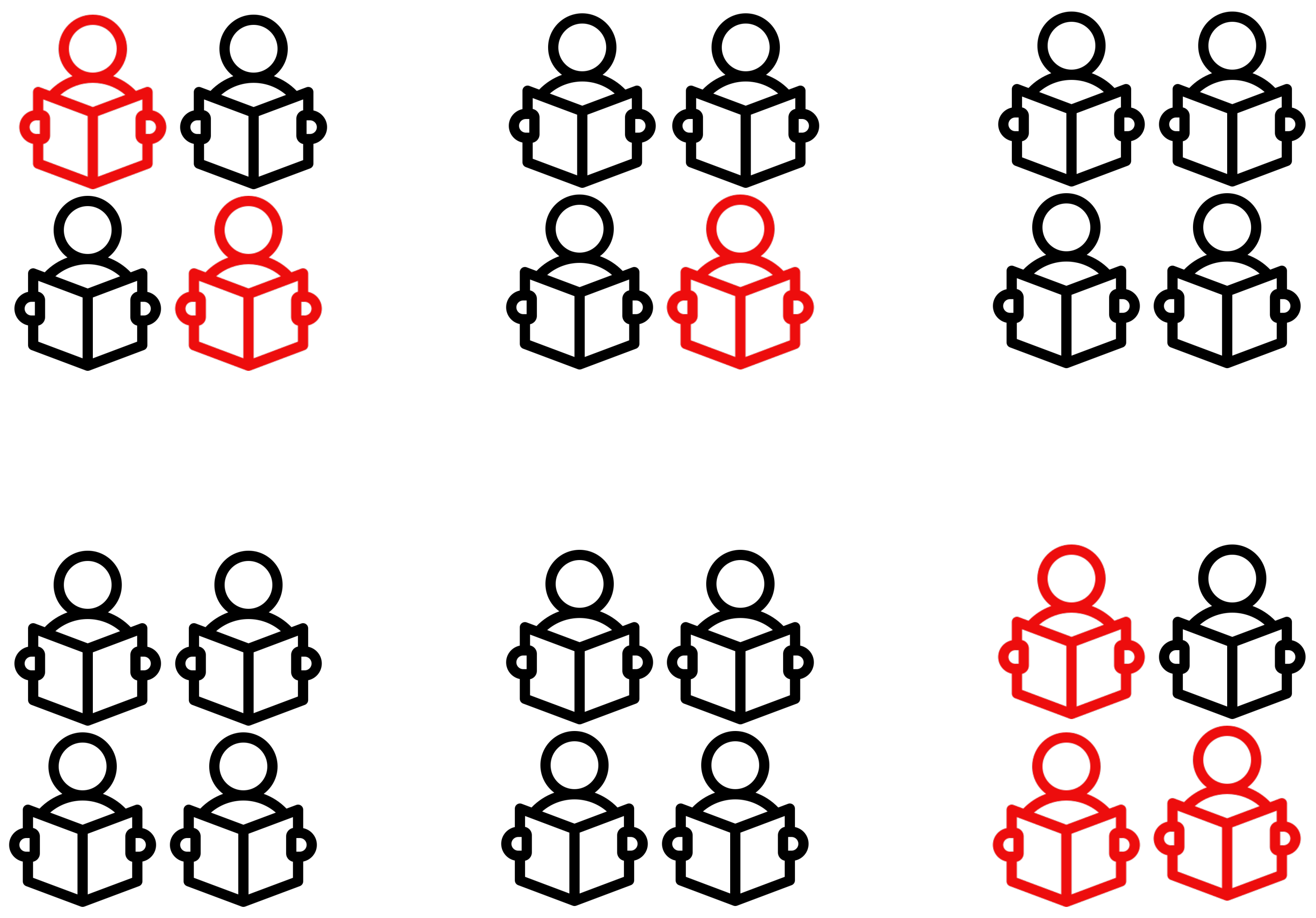
What if more classrooms are infected?



- This is the case $r \geq 2$
- We can have $(\mathscr{E}, r) - ECFF(t, n)$

The school problem

What if more classrooms are infected?



- This is the case $r \geq 2$
- We can have $(\mathscr{E}, r) - ECFF(t, n)$
- And also $(\mathscr{E}, r) - CFF(t, n)$

Generalizing the idea.. further

The edge-identifying CFF, $r \geq 2$

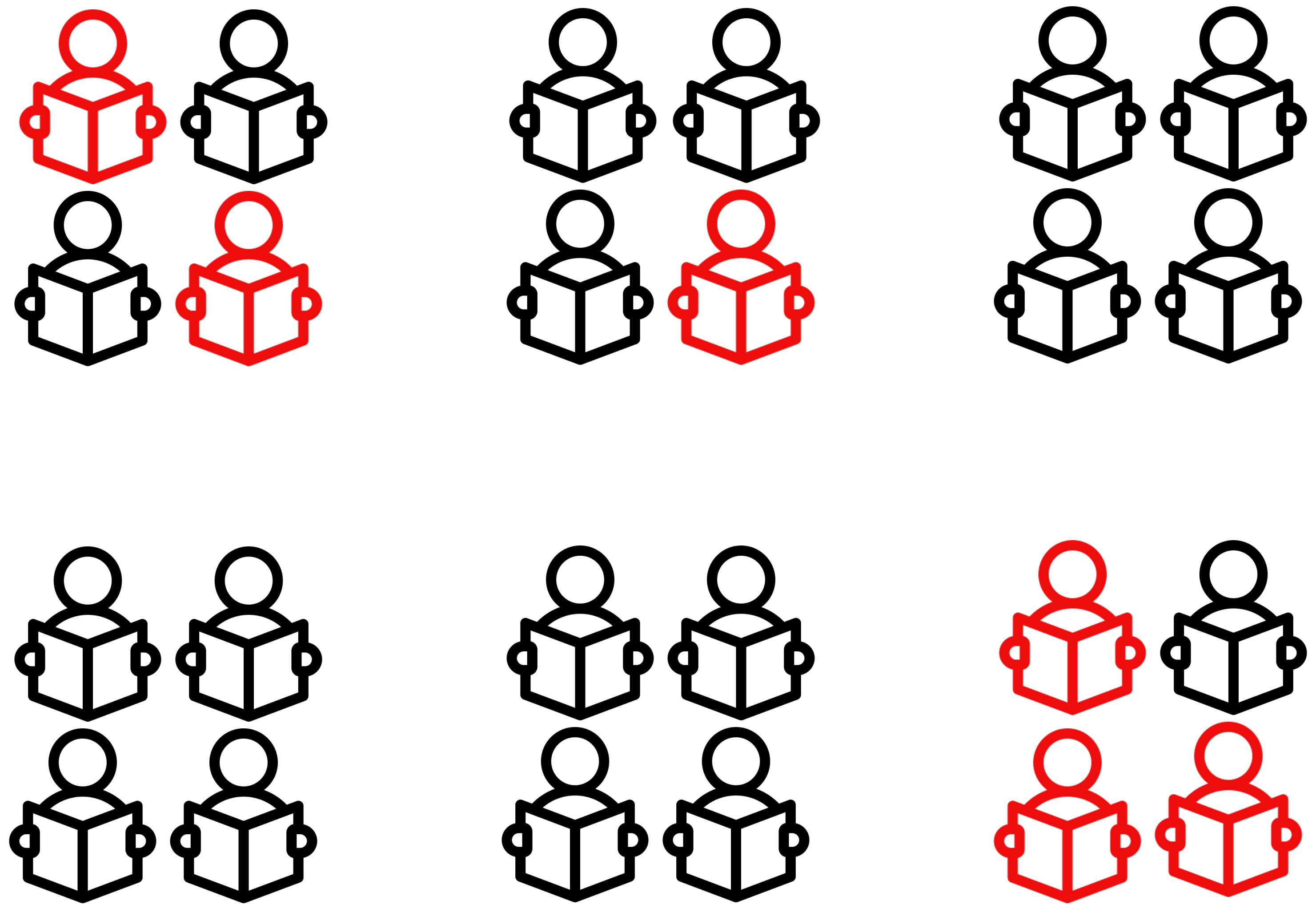
- Let $\mathscr{H} = ([1,n], \mathscr{E})$ be a hypergraph with *m* disjoint edges of cardinality *k*
- Construct $(\mathscr{E}, r) - ECFF(t, n)$ as follows:
  - **Step 1**: pick a good $r - CFF(t, m)$
  - **Step 2**: repeat each column *k* times
- We have $t_E(r, \mathscr{E}) = t(r, m)$

e_1	e_2	e_3	e_4	e_5	e_6
1	1	1	0	0	0
1	0	0	1	1	0
0	1	0	1	0	1
0	0	1	0	1	1

1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1

$(\mathscr{E}, r) - CFF(t, n)$

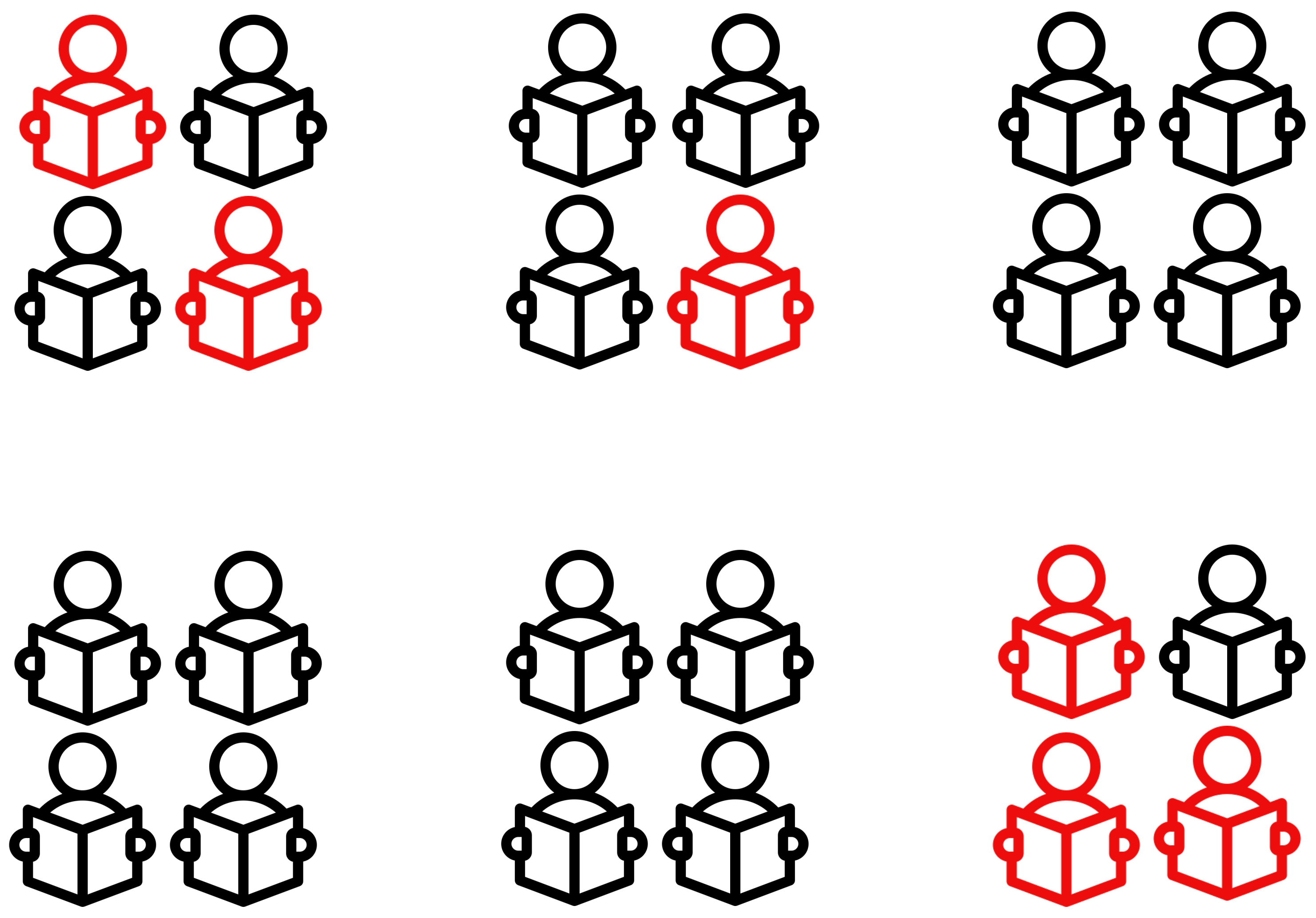
The vertex-identifying CFF, $r \geq 2$



- Propose some constructions of $(\mathscr{E}, r) - CFF$
  - For *m* classrooms of *k* students each
  - Identify r infected classrooms and everyone inside them

$(\mathscr{E}, r) - CFF(t, n)$

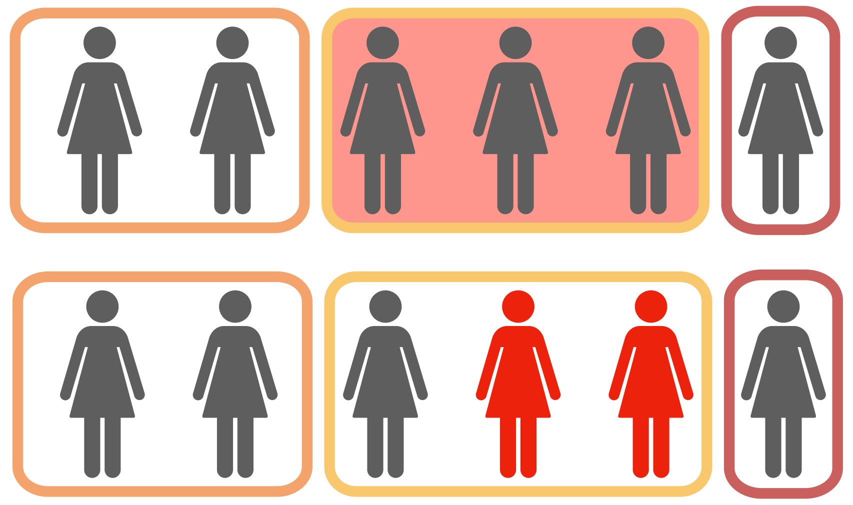
The vertex-identifying CFF, $r \geq 2$



- Propose some constructions of $(\mathscr{E}, r) - CFF$
  - For m classrooms of k students each
  - Identify r infected classrooms and everyone inside them
- Generalization of Li, van Rees and Wei (2006)
  - Uses an $r - CFF(t, m)$ and $(r-1) - CFF(t, m)$ to build $(\mathscr{E}, r) - CFF(kt, km)$
- Allows edges of different cardinalities

Bounds for CFFs on hypergraphs

Non-overlapping edges



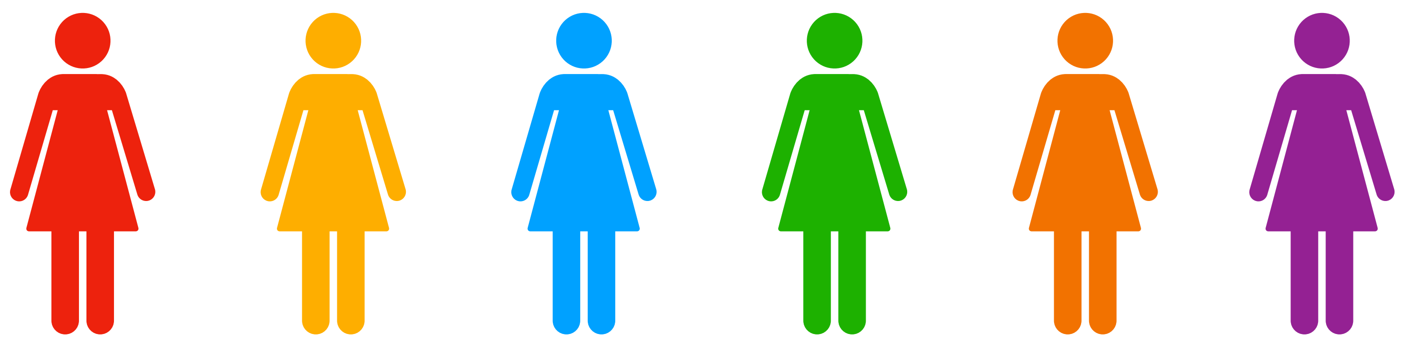
CFF	Lower Bound	Upper Bound
$(\mathcal{E},1)$-ECFF(t,n)	$\log(n/k)$	$\log(n/k)$
$(\mathcal{E},1)$-CFF(t,n)	$\log(n/k)$	$\log(n/k)+k$
$(\mathcal{E},r)$-ECFF(t,n)	$c_1 \frac{r^2\log(n/k)}{\log r}$	$c_2 \cdot r^2 \log(n/k)$
$(\mathcal{E},r)$-CFF(t,n)	$c_1 \frac{r^2\log(n/k)}{\log r}$	$c_3 \cdot k \cdot r^2 \log(n/k)$

$n = m \times k$

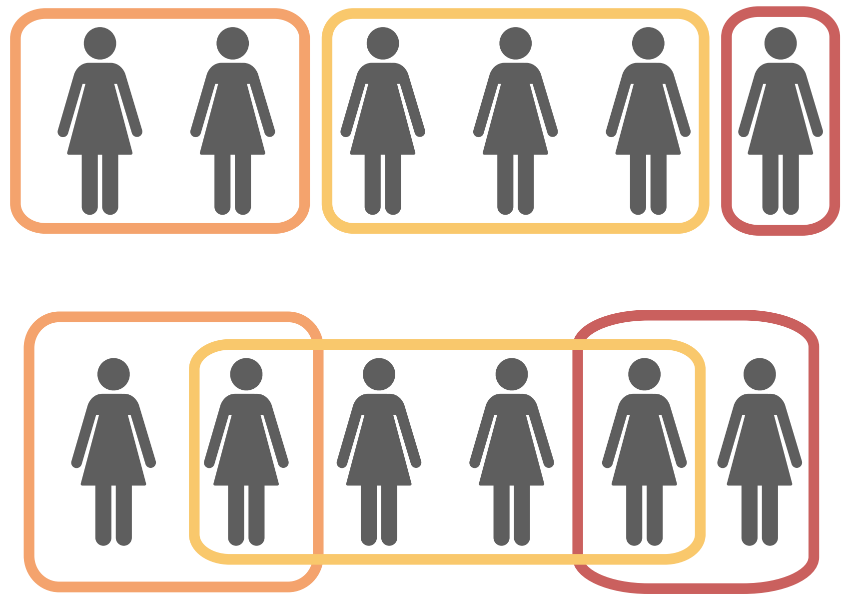
m edges with k vertices each

In this talk

1	1	1	0	0	0
1	0	0	1	1	0
0	1	0	1	0	1
0	0	1	0	1	1



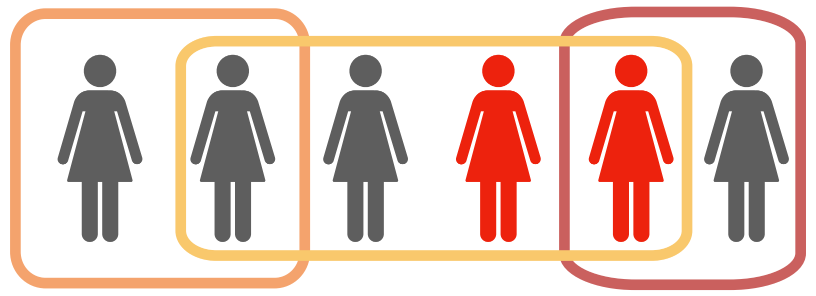
- The problem of **combinatorial group testing** in pandemic screening
- Study of ***CFFs on hypergraphs***
  - Non-overlapping edges
  - Overlapping edges
  - Some other cases
- Applications in cryptography



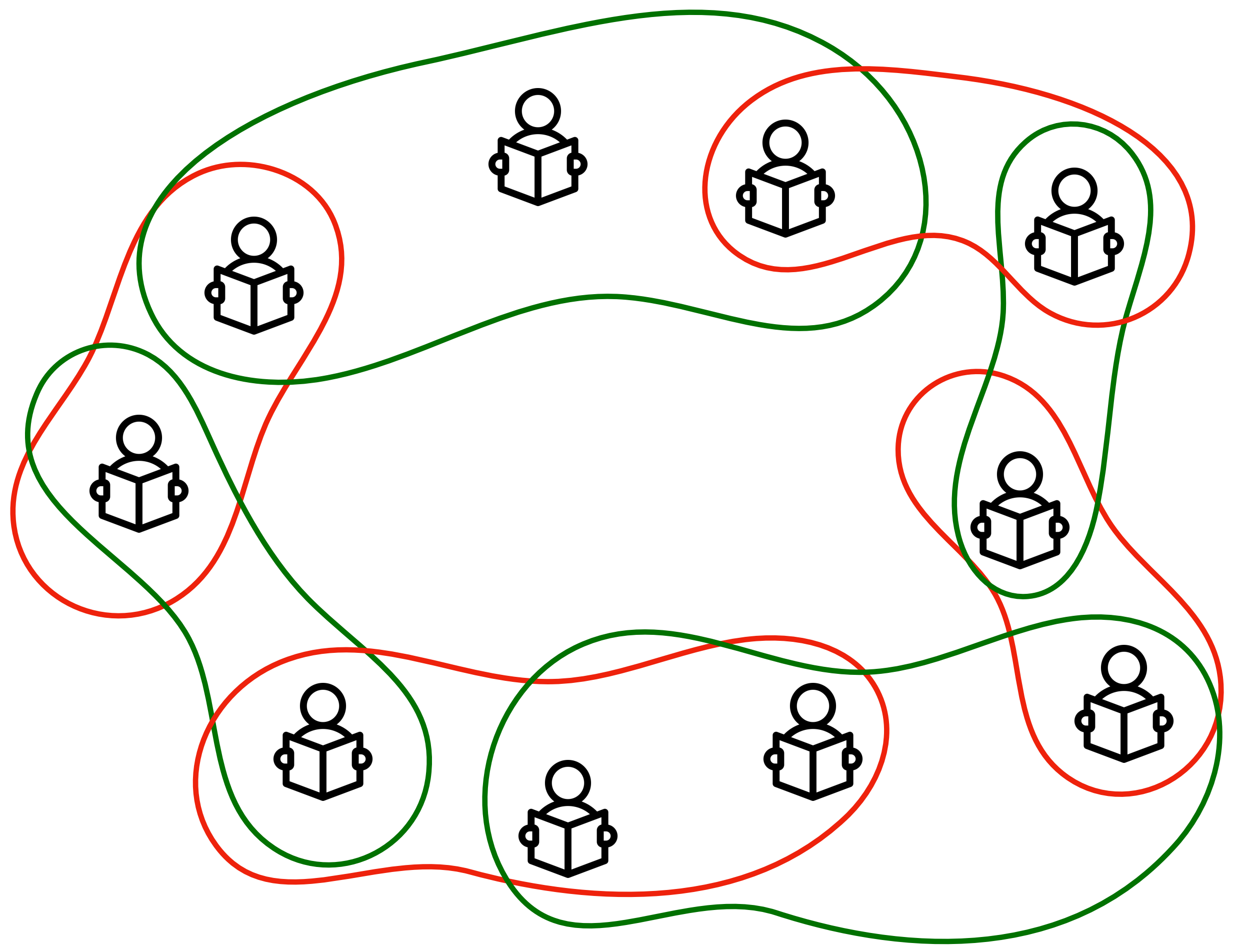
1. Bardini Idalino, T., Moura, L. Structure-Aware Combinatorial Group Testing: A New Method for PandemicScreening. In: IWOCA 2022. Lecture Notes in Computer Science, vol 13270. Springer, Cham. https://doi.org/10.1007/978-3-031-06678-8

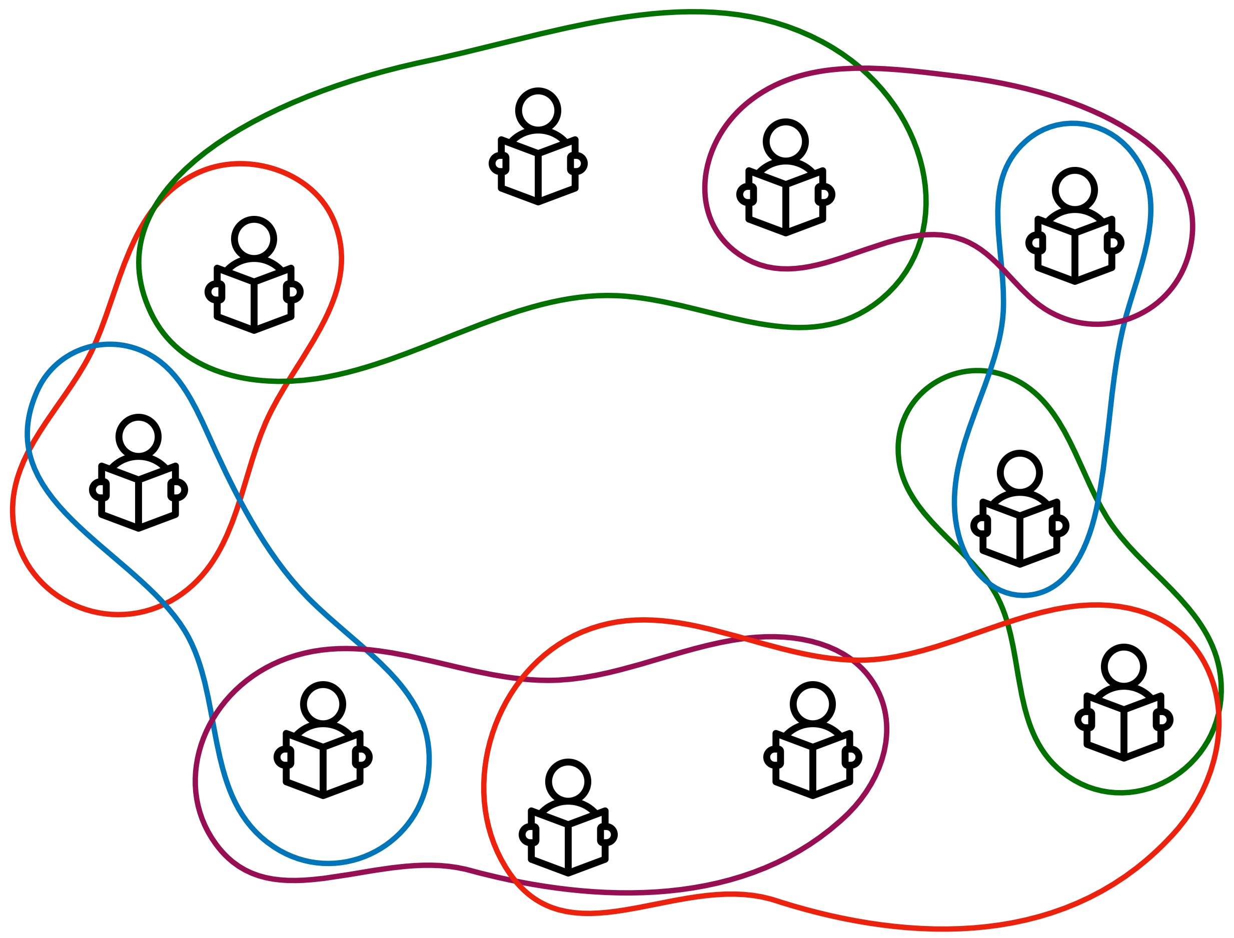
2. Bardini Idalino, T., Moura, L. Combinatorial group testing and cover-free families on hypergraphs. 2025[Unpublished manuscript].

Overlapping edges

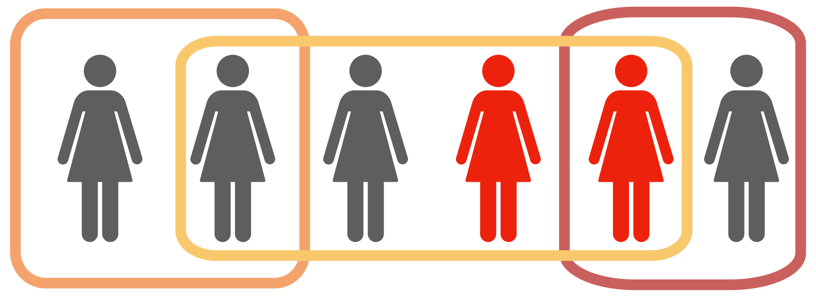


- Propose constructions inspired by the non-overlapping ones
- Use **edge-colouring** to partition the edges into sets of non-overlapping edges
  - Edge-colouring: no vertex is incident to more than one edge of the same colour
  - Strong edge-coloring: any two vertices belonging to distinct edges with the same colour are not adjacent.





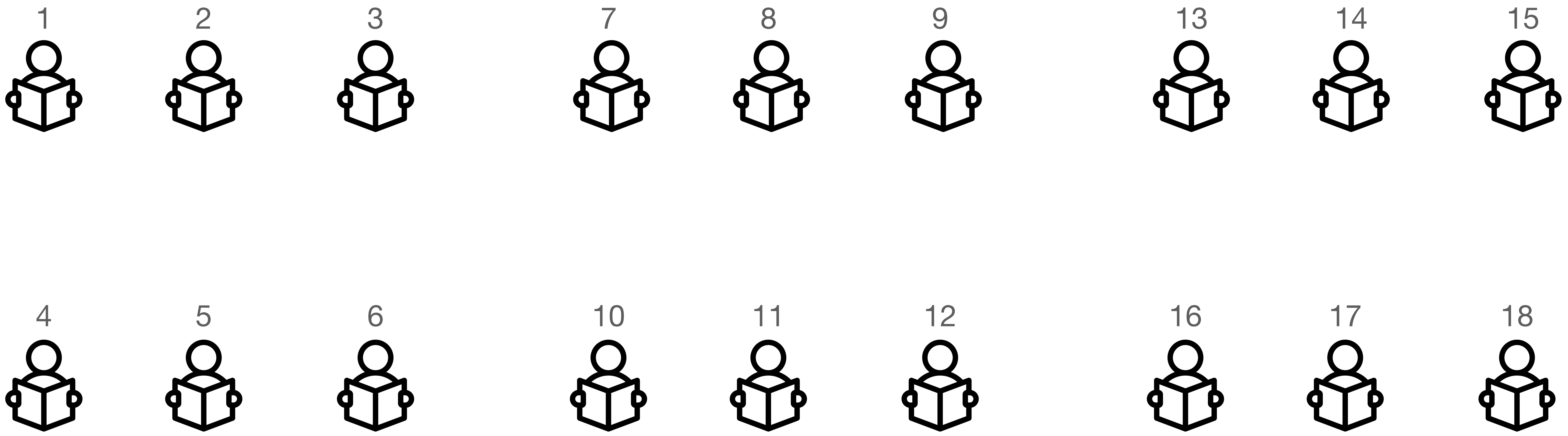
Overlapping edges



- Construction of $(\mathscr{E},1) - CFF$ and $(\mathscr{E},1) - ECFF$ based on edge-colouring
- Construction of $(\mathscr{E}, r) - CFF$ based on strong edge-colouring
- ***Defect cover***: a set of at most r edges whose union contains the set of infected elements
  - We can handle many infected edges, as long as the size of the defect cover is $\leq r$

The high school problem

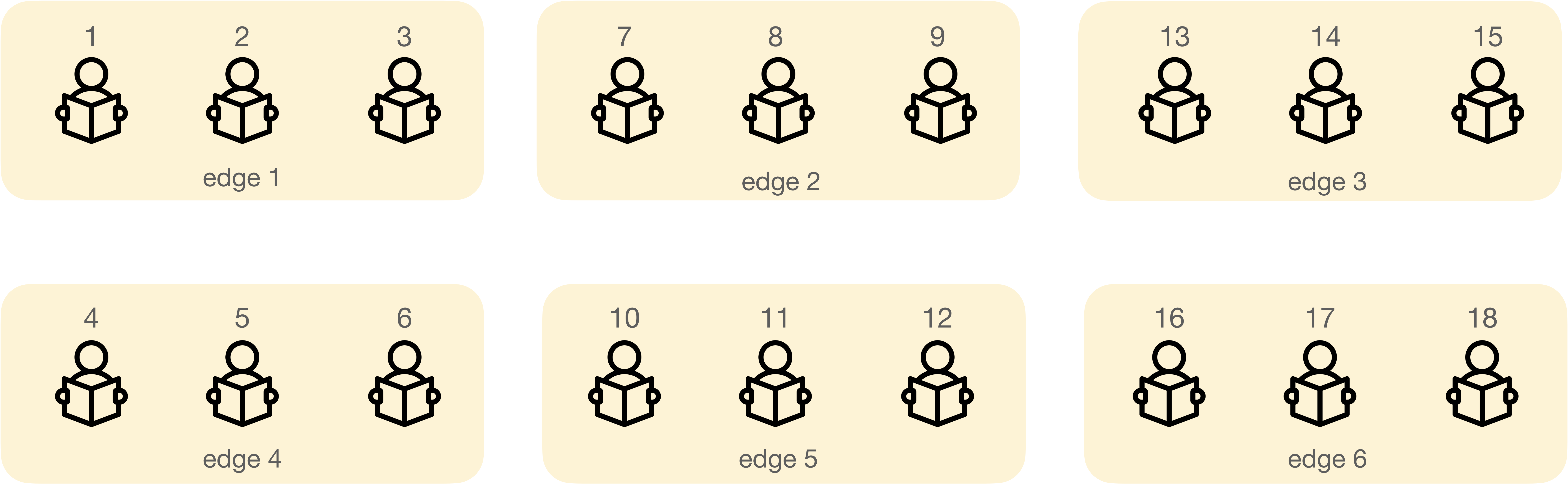
$r = 1$



The high school problem

$r = 1$

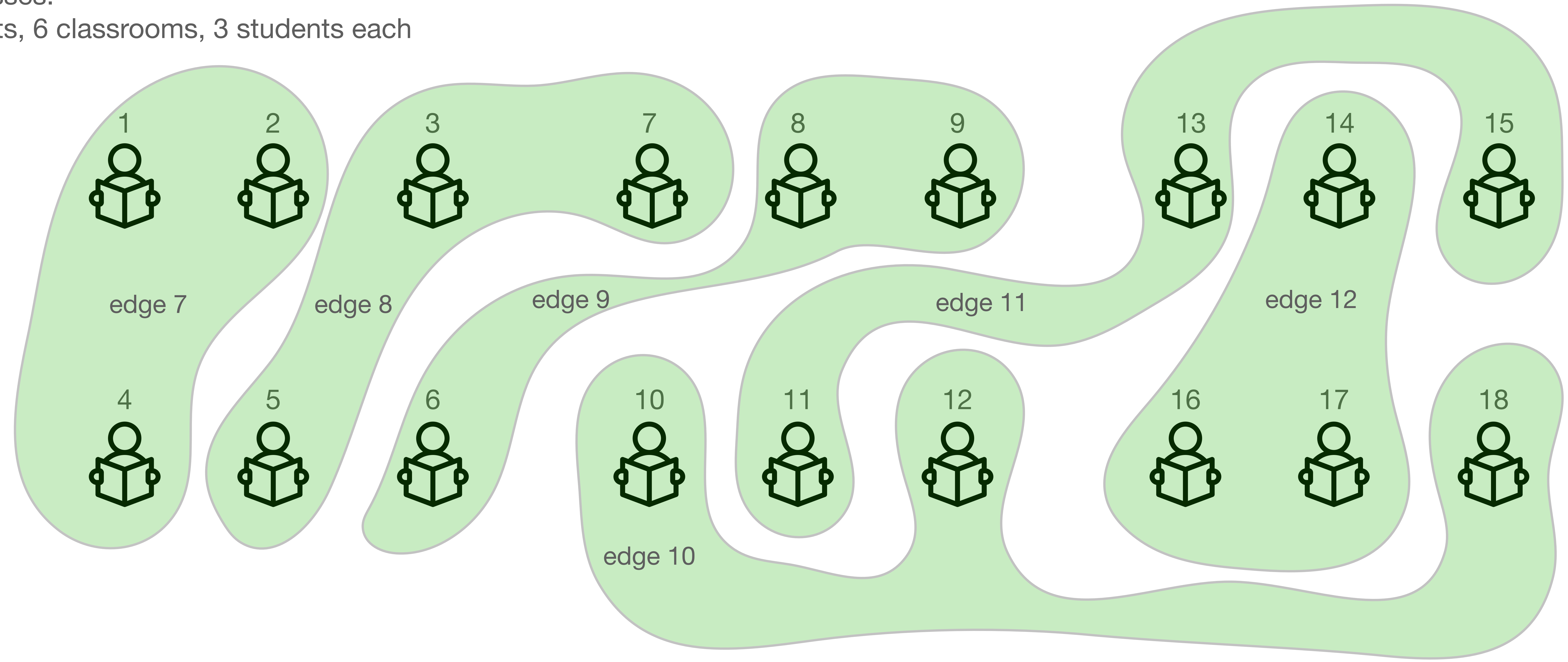
Morning classes:
n = 18 students, 6 classrooms, 3 students each



The high school problem

$r = 1$

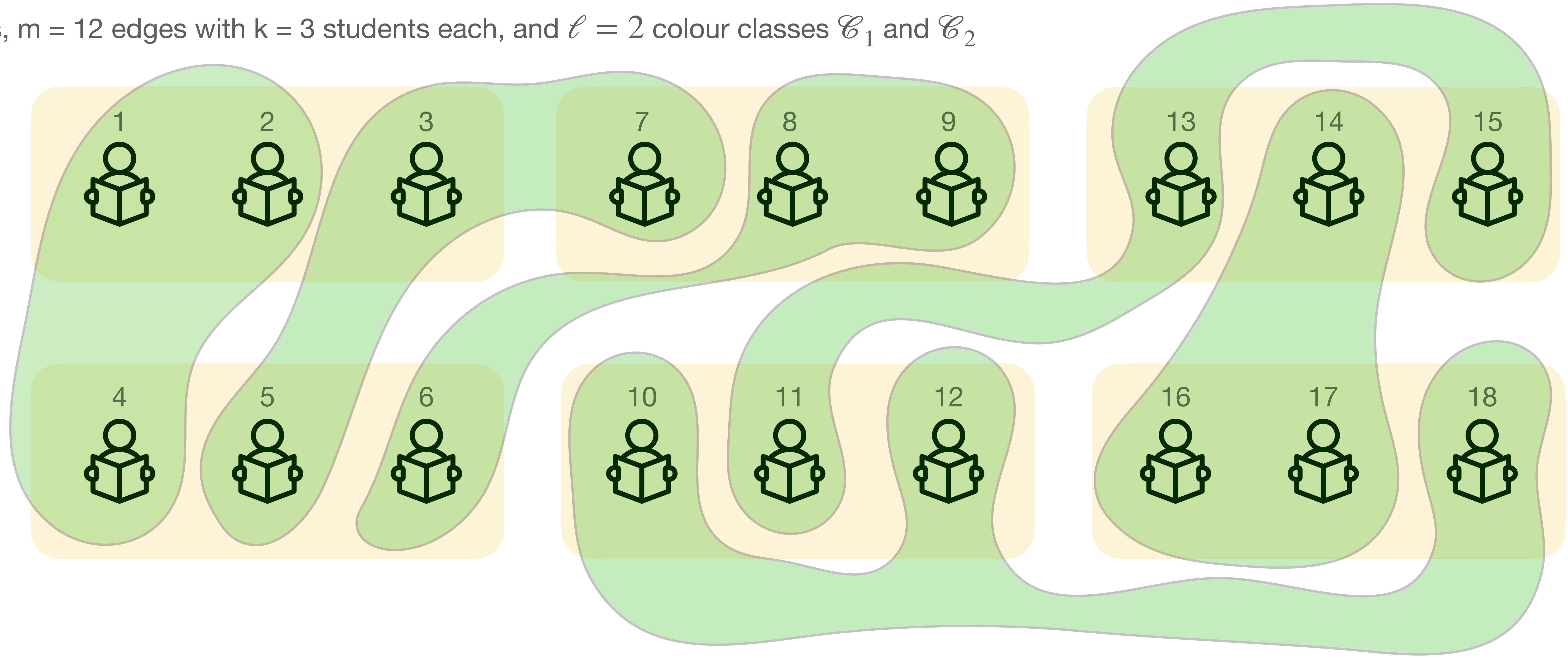
Afternoon classes:
n = 18 students, 6 classrooms, 3 students each



The high school problem

$r = 1$

Total:
n = 18 vertices, m = 12 edges with k = 3 students each, and $\ell = 2$ colour classes $\mathscr{C}_1$ and $\mathscr{C}_2$



Overlapping edge construction

$\mathscr{C}_1$

edge 1	edge 2	edge 3	edge 4	edge 5	edge 6
1	1	1	0	0	0
1	0	0	1	1	0
0	1	0	1	0	1
0	0	1	0	1	1

$\mathscr{C}_2$

edge 7	edge 8	edge 9	edge 10	edge 11	edge 12
1	1	1	0	0	0
1	0	0	1	1	0
0	1	0	1	0	1
0	0	1	0	1	1

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0
0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1
0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	1	1	1	1	0	1	0	0	1
0	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	1	1
0	0	0	0	0	1	0	1	1	0	1	0	1	1	1	1	1	0

Overlapping edge construction

$\mathscr{C}_1$

edge 1	edge 2	edge 3	edge 4	edge 5	edge 6
1	1	1	0	0	0
1	0	0	1	1	0
0	1	0	1	0	1
0	0	1	0	1	1

$\mathscr{C}_2$

edge 7	edge 8	edge 9	edge 10	edge 11	edge 12
1	1	1	0	0	0
1	0	0	1	1	0
0	1	0	1	0	1
0	0	1	0	1	1

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0
0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1
0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	1	1	1	1	0	1	0	0	1
0	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	1	1
0	0	0	0	0	1	0	1	1	0	1	0	1	1	1	1	1	0
1	0	1	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0
0	1	0	0	1	0	0	1	0	0	0	1	1	0	0	1	0	0
0	0	0	1	0	0	1	0	1	0	0	0	0	0	1	0	1	1

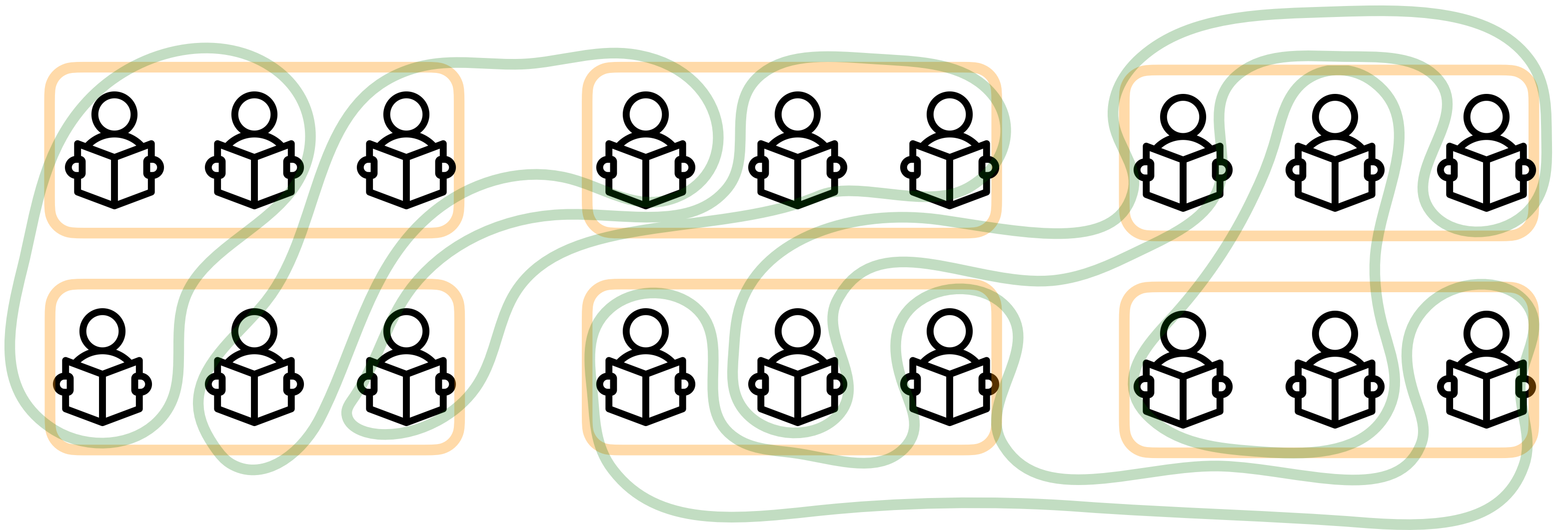
Overlapping edge construction

$r = 1$

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0
0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1
0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1

1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	1	1	1	1	0	1	0	0	1
0	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	1	1
0	0	0	0	0	1	0	1	1	0	1	0	1	1	1	1	1	0
1	0	1	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0
0	1	0	0	1	0	0	1	0	0	0	1	1	0	0	1	0	0
0	0	0	1	0	0	1	0	1	0	0	0	0	0	1	0	1	1

$$(\mathscr{E},1) - CFF(t,n)$$



Overlapping edge construction

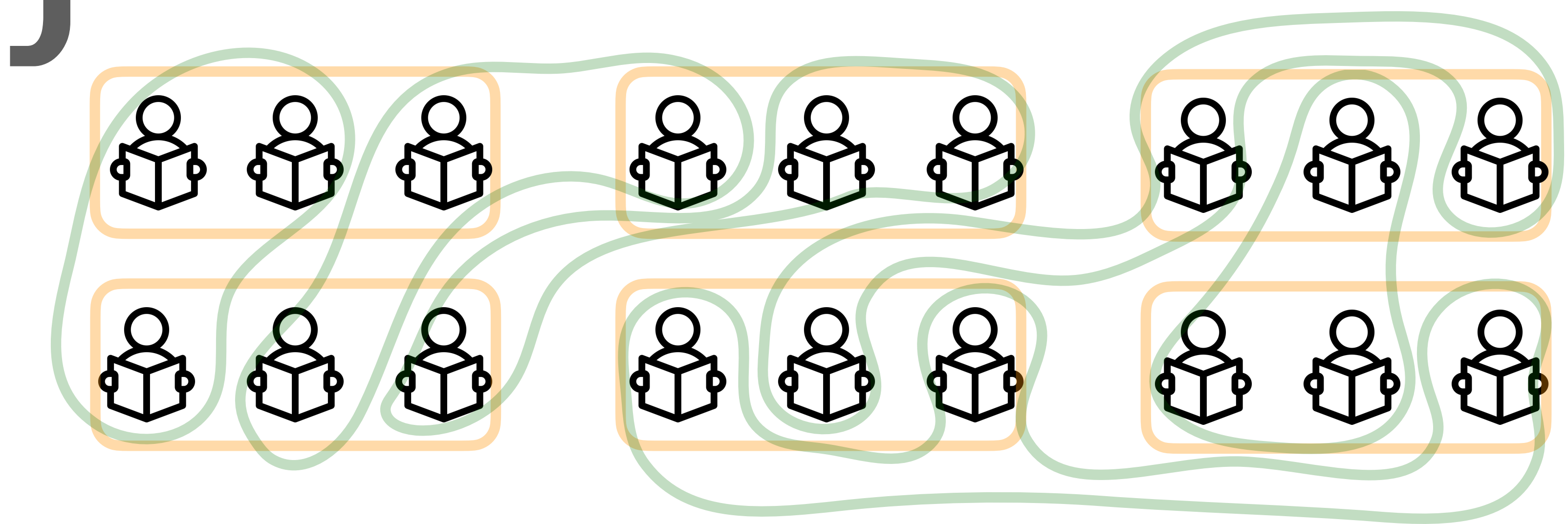
$r = 1$

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0
0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1
0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1

1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	1	1	1	1	0	1	0	0	1
0	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	1	1
0	0	0	0	0	1	0	1	1	0	1	0	1	1	1	1	1	0
1	0	1	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0
0	1	0	0	1	0	0	1	0	0	0	1	1	0	0	1	0	0
0	0	0	1	0	0	1	0	1	0	0	0	0	0	1	0	1	1

Identify the infected edge

Inside an infected edge, we can identify precisely the infected students



$(\mathscr{E},1) - CFF(t, n)$

Overlapping edge construction

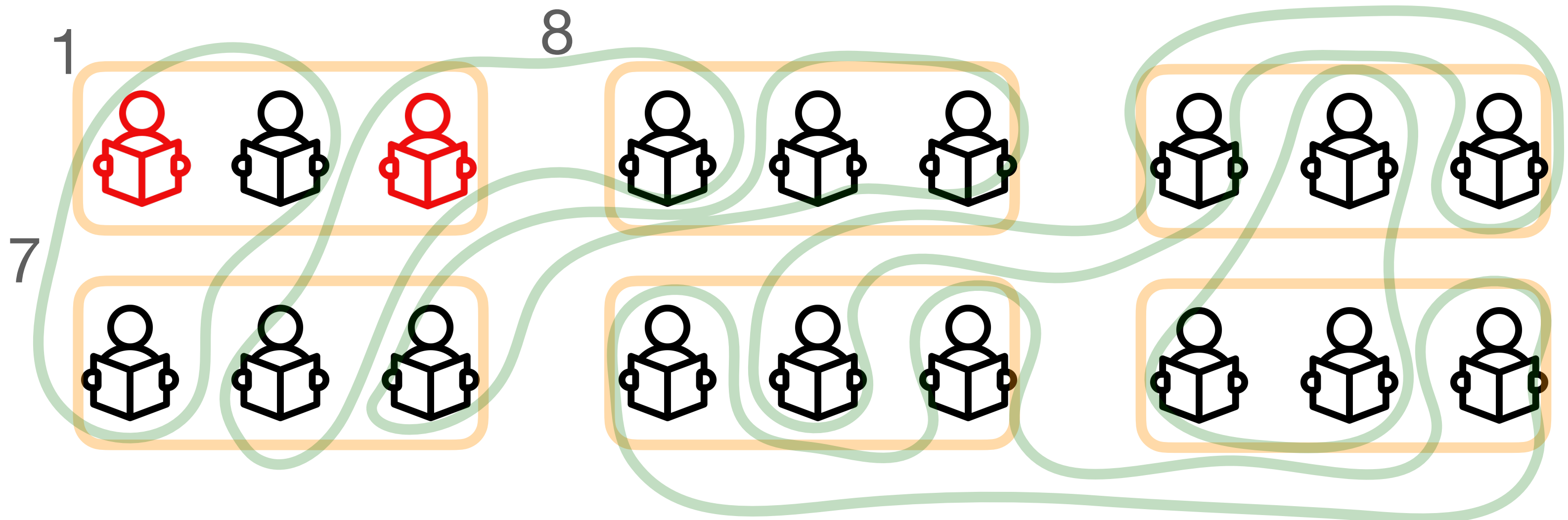
$r = 1$

Edges 1, 7 and 8 are infected

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0
0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1
0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1

1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	1	1	1	1	0	1	0	0	1
0	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	1	1
0	0	0	0	0	1	0	1	1	0	1	0	1	1	1	1	1	0
1	0	1	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0
0	1	0	0	1	0	0	1	0	0	0	1	1	0	0	1	0	0
0	0	0	1	0	0	1	0	1	0	0	0	0	0	1	0	1	1

$(\mathscr{E}, 1) - CFF(t, n)$



Overlapping edge construction

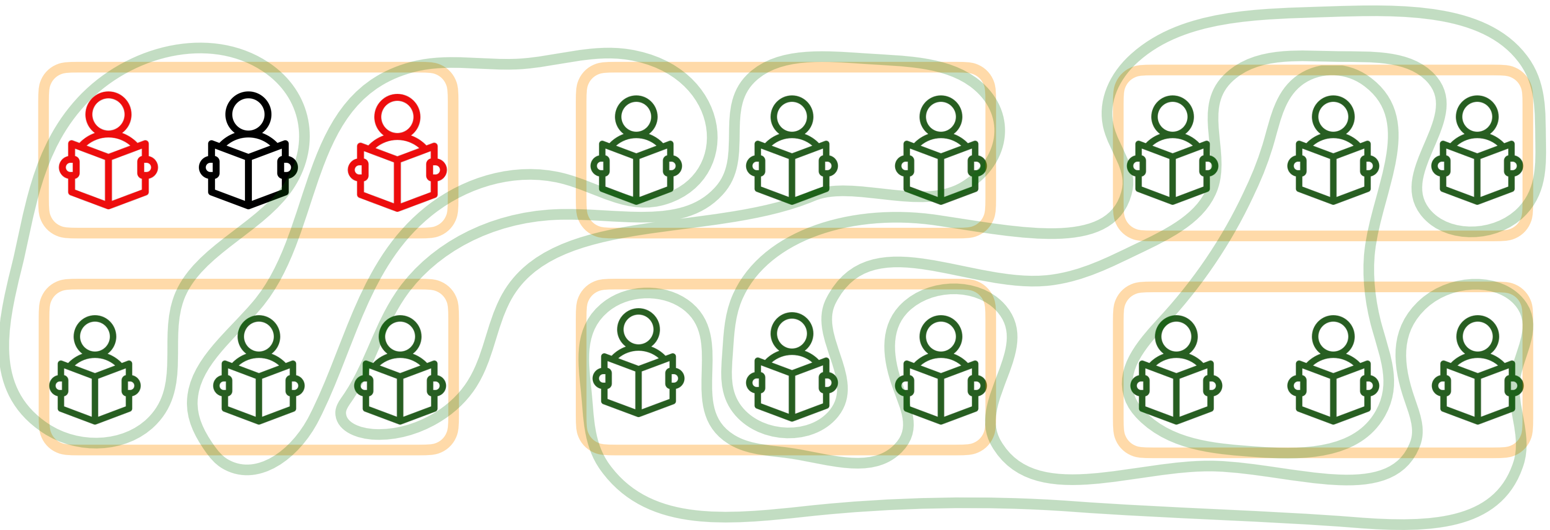
$r = 1$

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0
0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1
0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1

Edges 1, 7 and 8 are infected, students 4-18 are cleared out

1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	1	1	1	1	0	1	0	0	1
0	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	1	1
0	0	0	0	0	1	0	1	1	0	1	0	1	1	1	1	1	0
1	0	1	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0
0	1	0	0	1	0	0	1	0	0	0	1	1	0	0	1	0	0
0	0	0	1	0	0	1	0	1	0	0	0	0	0	1	0	1	1

$(\mathscr{E},1) - CFF(t,n)$



Overlapping edge construction

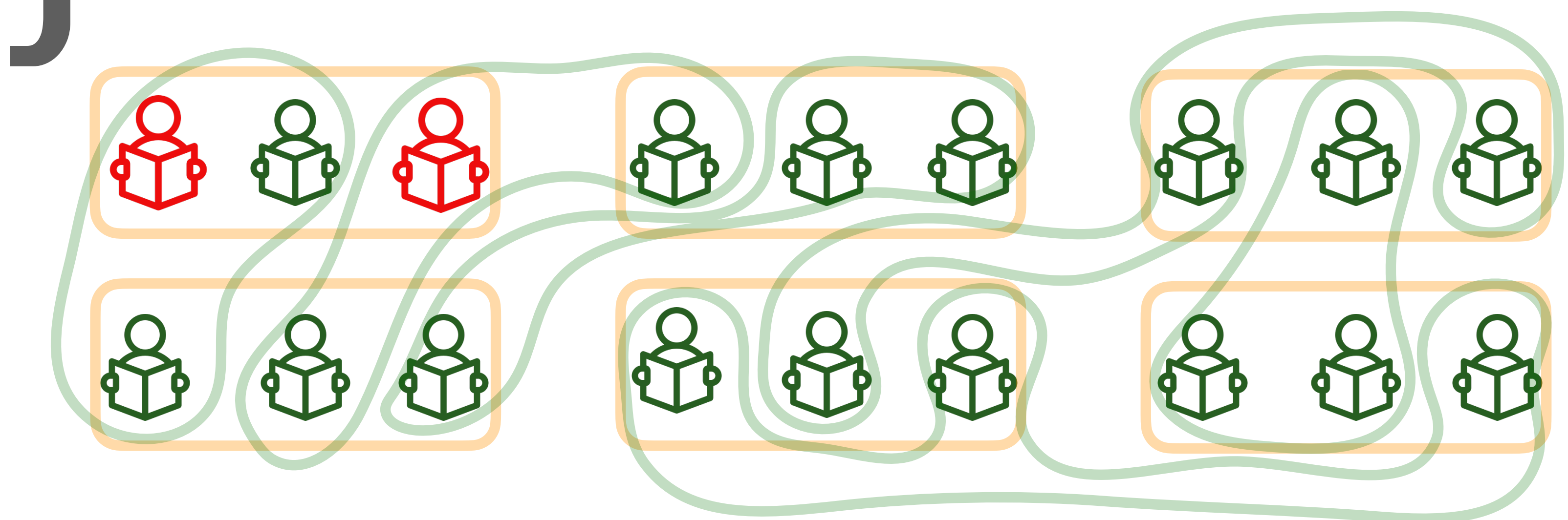
$r = 1$

Edges 1, 7 and 8 are infected, students 4-18 are cleared out

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0
0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1
0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1

1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	1	1	1	1	0	1	0	0	1
0	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	1	1
0	0	0	0	0	1	0	1	1	0	1	0	1	1	1	1	1	0
1	0	1	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0
0	1	0	0	1	0	0	1	0	0	0	1	1	0	0	1	0	0
0	0	0	1	0	0	1	0	1	0	0	0	0	0	1	0	1	1

Inside an infected edge, we can identify precisely the infected students



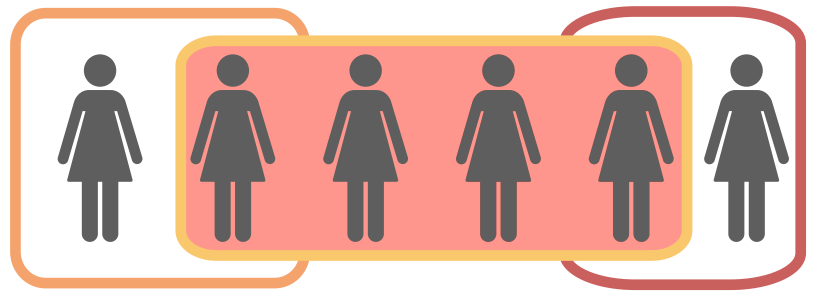
$$(\mathscr{E},1) - CFF(t = 2 \times (4+3), n = 18)$$

$$(\mathscr{E},1) - ECFF(t = 2 \times 4, n = 18)$$

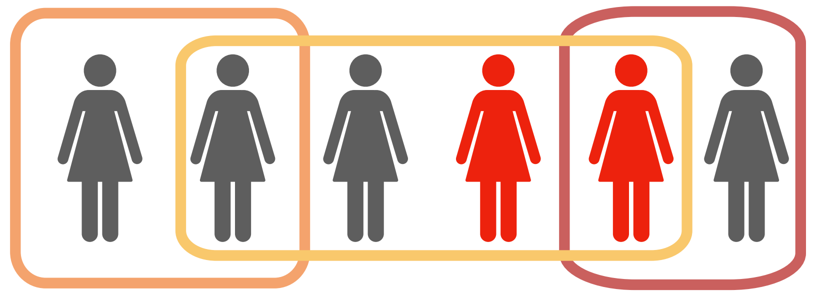
Overlapping edge construction

$r = 1$

- Let $\mathscr{H} = ([1,n], \mathscr{E})$ be a hypergraph with *m* edges of cardinality *k*
- And let $\chi(\mathscr{H}) = \ell$ be its edge chromatic number, with colour classes $\mathscr{C}_1, \ldots, \mathscr{C}_\ell$
- We constructed an $(\mathscr{E},1) - ECFF(t, n)$
  - $t_E(1,\mathscr{E}) \leq \ell \times t(1,m) \approx \ell \times \log m$



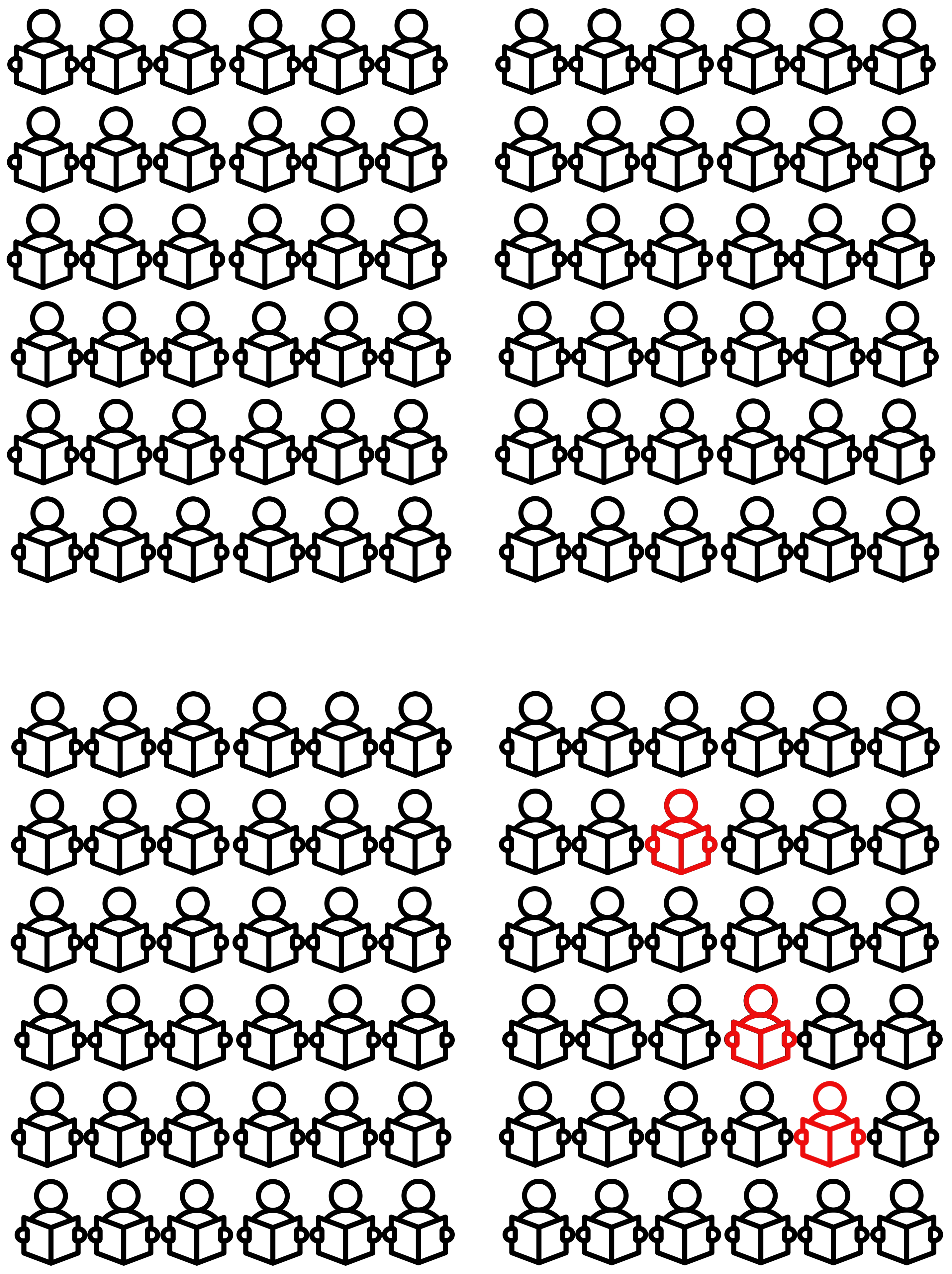
- We constructed a $(\mathscr{E},1) - CFF(t, n)$
  - $t(1,\mathscr{E}) \leq \ell \times (t(1,m) + k) \approx \ell \times (\log m + k)$



For a larger high school

$r = 1$

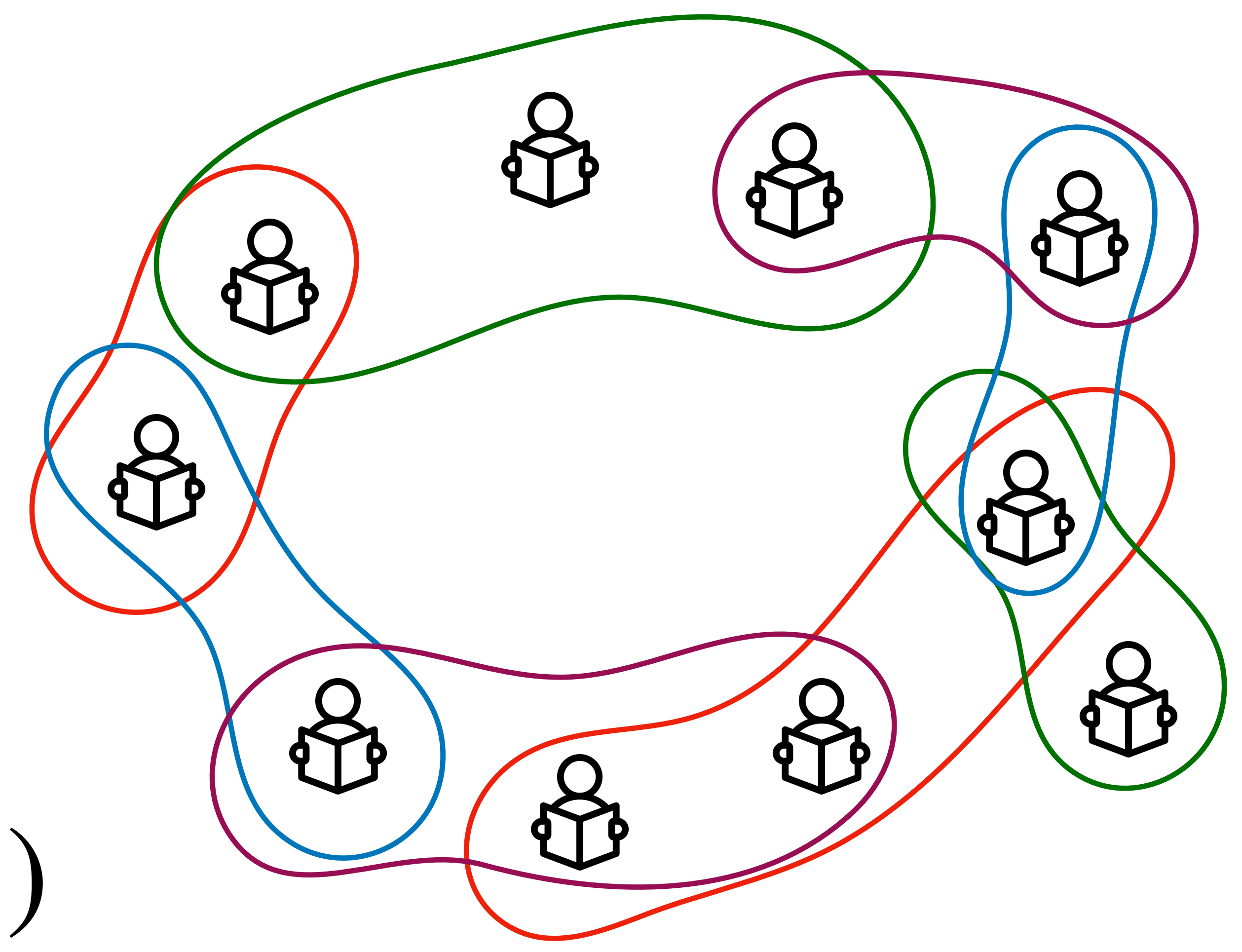
- Total of $n = 900$ students
- Each student takes $\ell = 4$ courses (4 colour classes)
- Each course has $k = 30$ students (cardinality of the edges)
- Total of $m = 120$ courses (edges)
  - Each color class has $120/4 = 30$ edges
- Construction:
  - Use $1 - CFF(7,30)$
  - $\ell \times t(1,m) = 4 \times 7 = 28$ tests to detect the infected edge
  - $\ell \times (t(1,m) + k) = 4 \times (7 + 30) = 148$ tests to identify all infected students in the infected edge



Overlapping edge construction

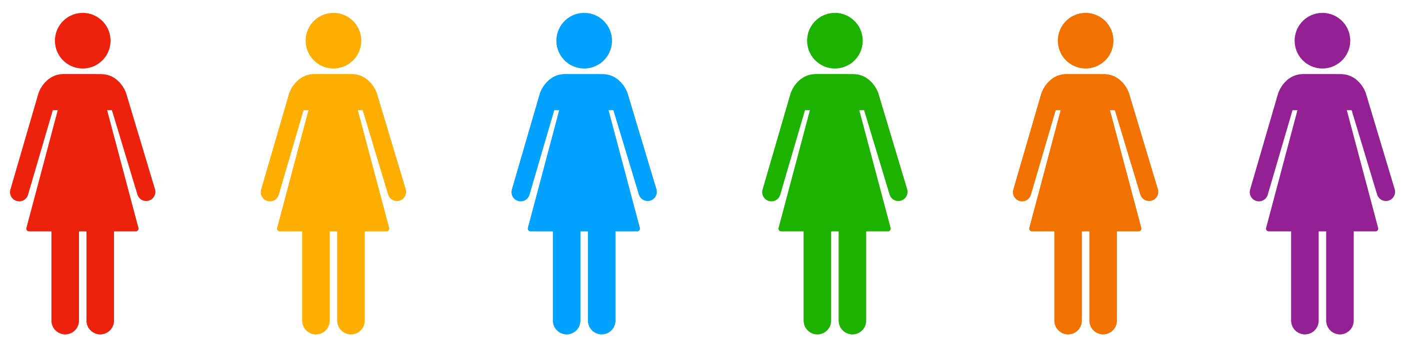
$r \geq 2$

- Generalization for $(\mathscr{E}, r) - CFF(t, n)$ using **strong edge-colouring**
  - At most r edges $I = \{e_1, e_2, \ldots, e_r\}$ contain all infected individuals
  - There are at most r edges in each $\mathscr{C}_i$ that intersect I
    - $\mathscr{C}_i$ contains at most r infected edges
- Use a combination of $r - CFF(t, |\mathscr{C}_i|)$ and $(r-1) - CFF(t', |\mathscr{C}_i|)$
  - $(\mathscr{E}, r) - CFF(t, n)$ with $t \leq \sum_{i=1}^{\ell} (t_i + k_i t'_i)$, $k_i =$ max edge in colour class $\mathscr{C}_i$

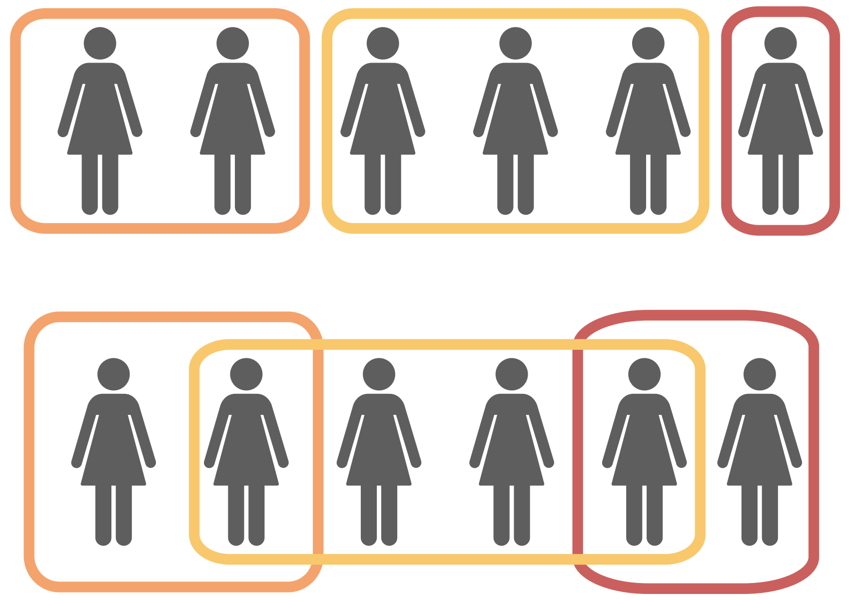


In this talk

1	1	1	0	0	0
1	0	0	1	1	0
0	1	0	1	0	1
0	0	1	0	1	1



- The problem of **combinatorial group testing** in pandemic screening
- Study of ***CFFs on hypergraphs***
  - Non-overlapping edges
  - Overlapping edges
  - Some other cases
- Applications in cryptography

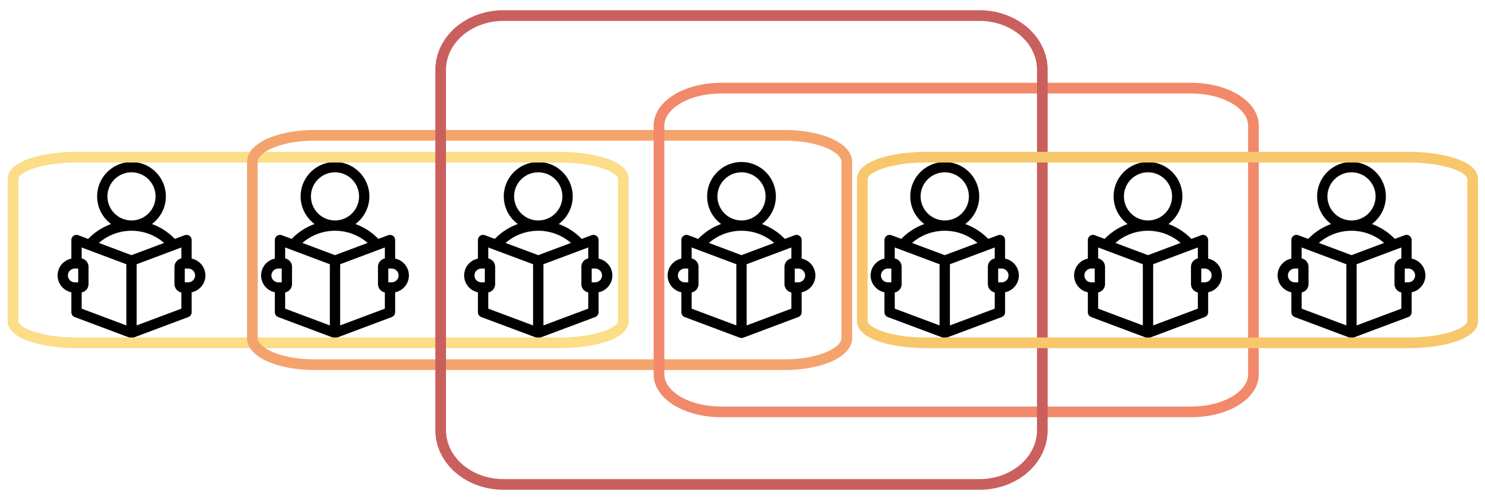


1. Bardini Idalino, T., Moura, L. Structure-Aware Combinatorial Group Testing: A New Method for PandemicScreening. In: IWOCA 2022. Lecture Notes in Computer Science, vol 13270. Springer, Cham. https://doi.org/10.1007/978-3-031-06678-8
2. Bardini Idalino, T., Moura, L. Combinatorial group testing and cover-free families on hypergraphs. 2025[Unpublished manuscript].

Some other cases

Consecutive infections

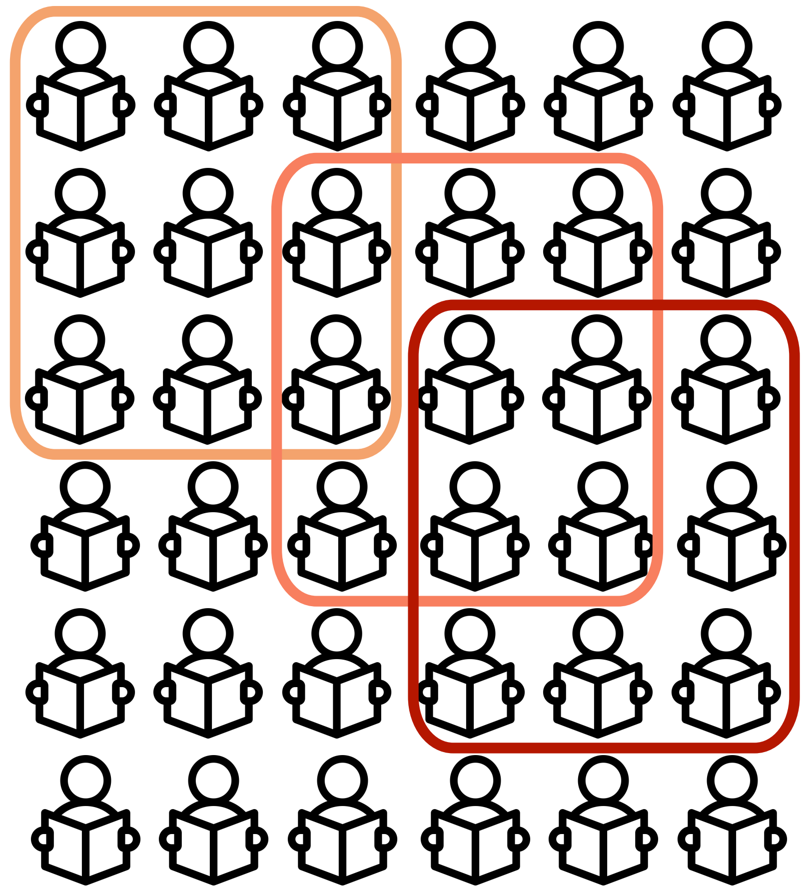
- In some scenarios, infections happen with people who are seated close to each other



Some other cases

Consecutive infections

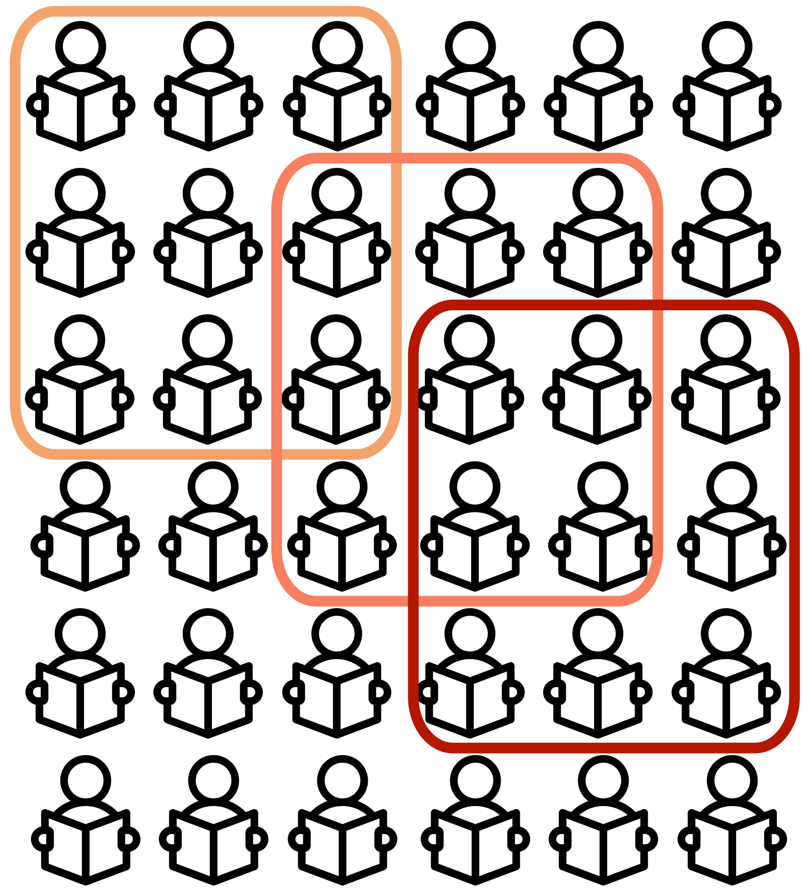
- In some scenarios, infections happen with people who are seated close to each other



Some other cases

Consecutive infections

- In some scenarios, infections happen with people who are seated close to each other
- We consider *consecutive hypercube hypergraphs*
- Initial results on CFFs for this case using vertex colouring



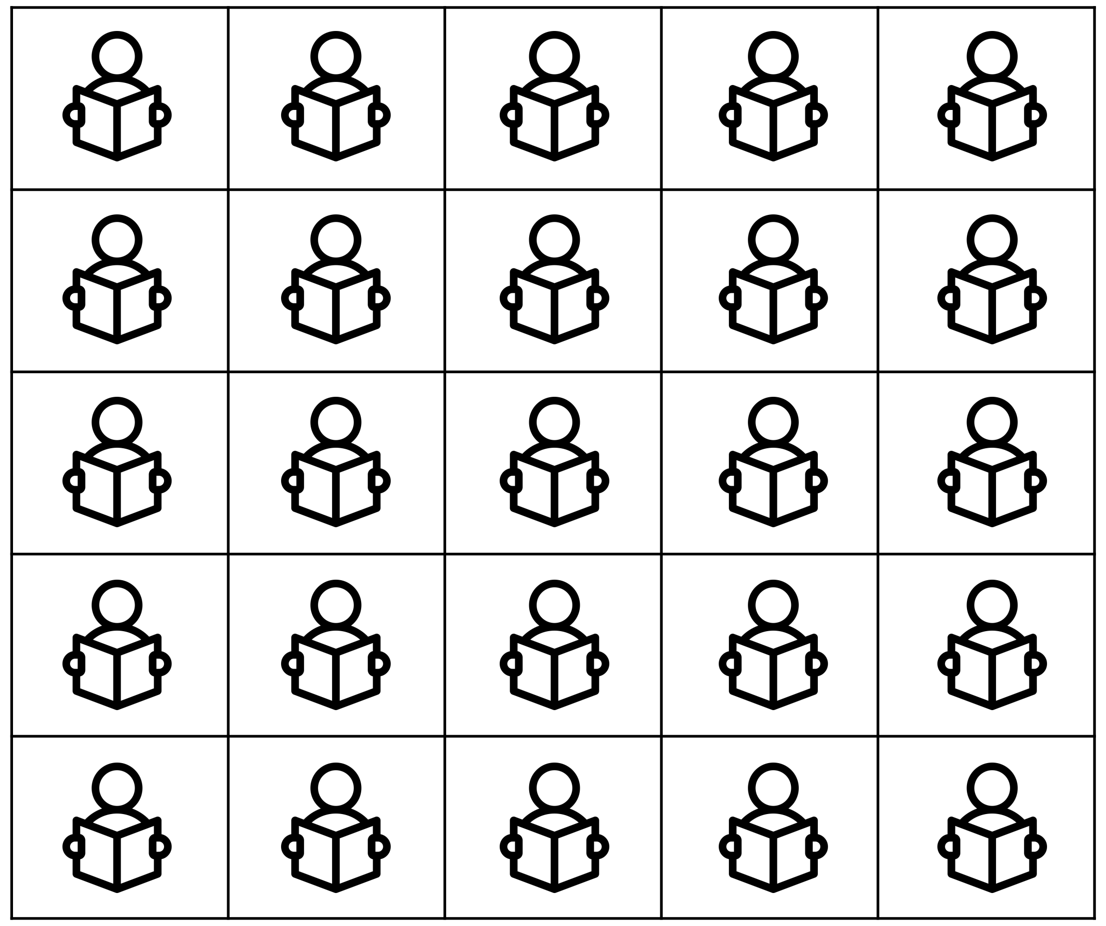
Some other cases

The Swiss-Army-Knife CFF

- In some scenarios, we don’t know which CFF to use
- We focus on constructions for CFFs on hypergraphs that hold *several purposes*
- For example, a matrix can be at the same time:
  - an $(\mathscr{E}, r^+) - ECFF(t, n)$
  - a $(\mathscr{E}, r^-) - CFF(t, n)$
  - A traditional $d - CFF(t, n)$
  - $r^+ > r^- > d$

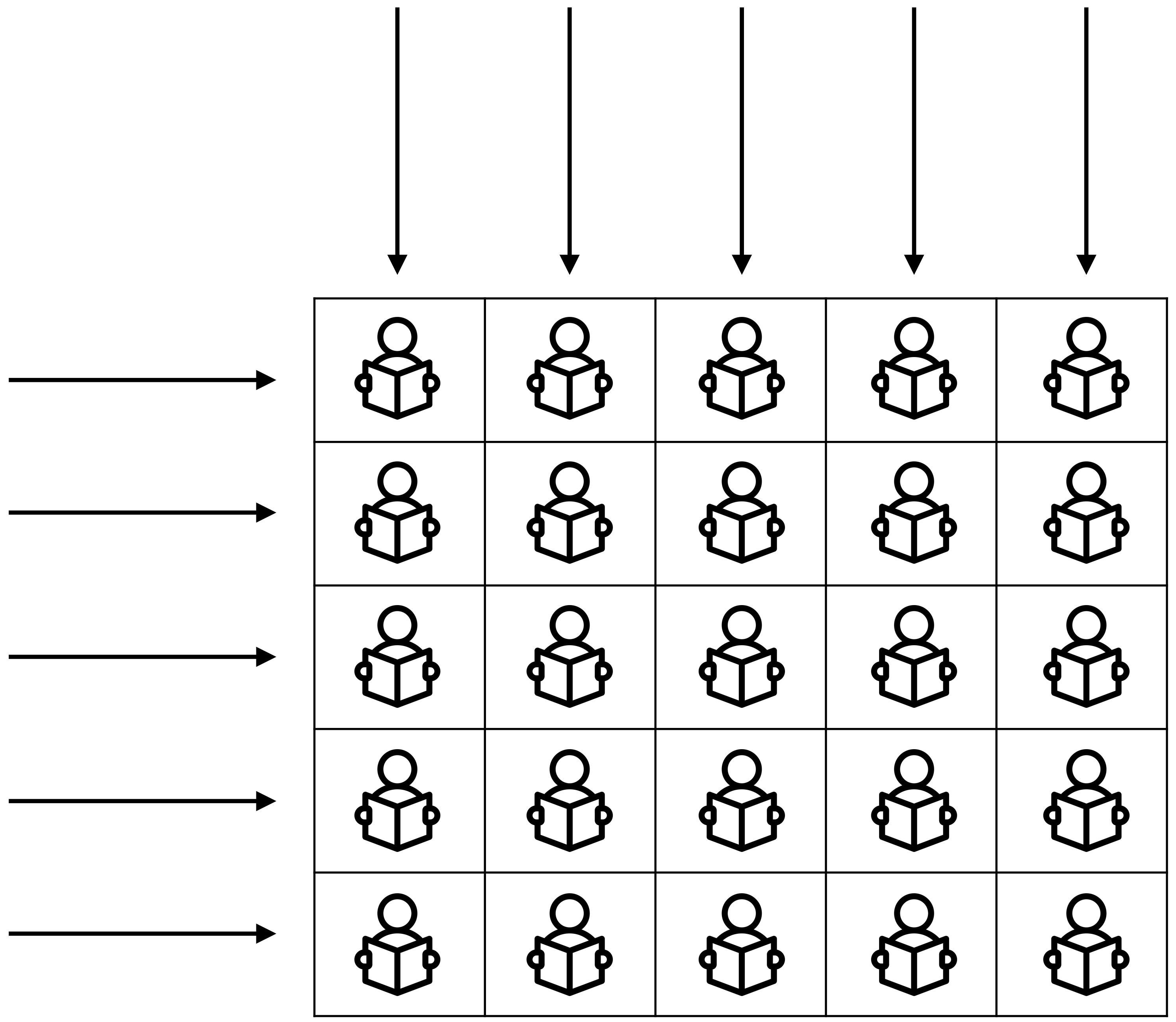
Some other cases

The Swiss-Army-Knife CFF



Some other cases

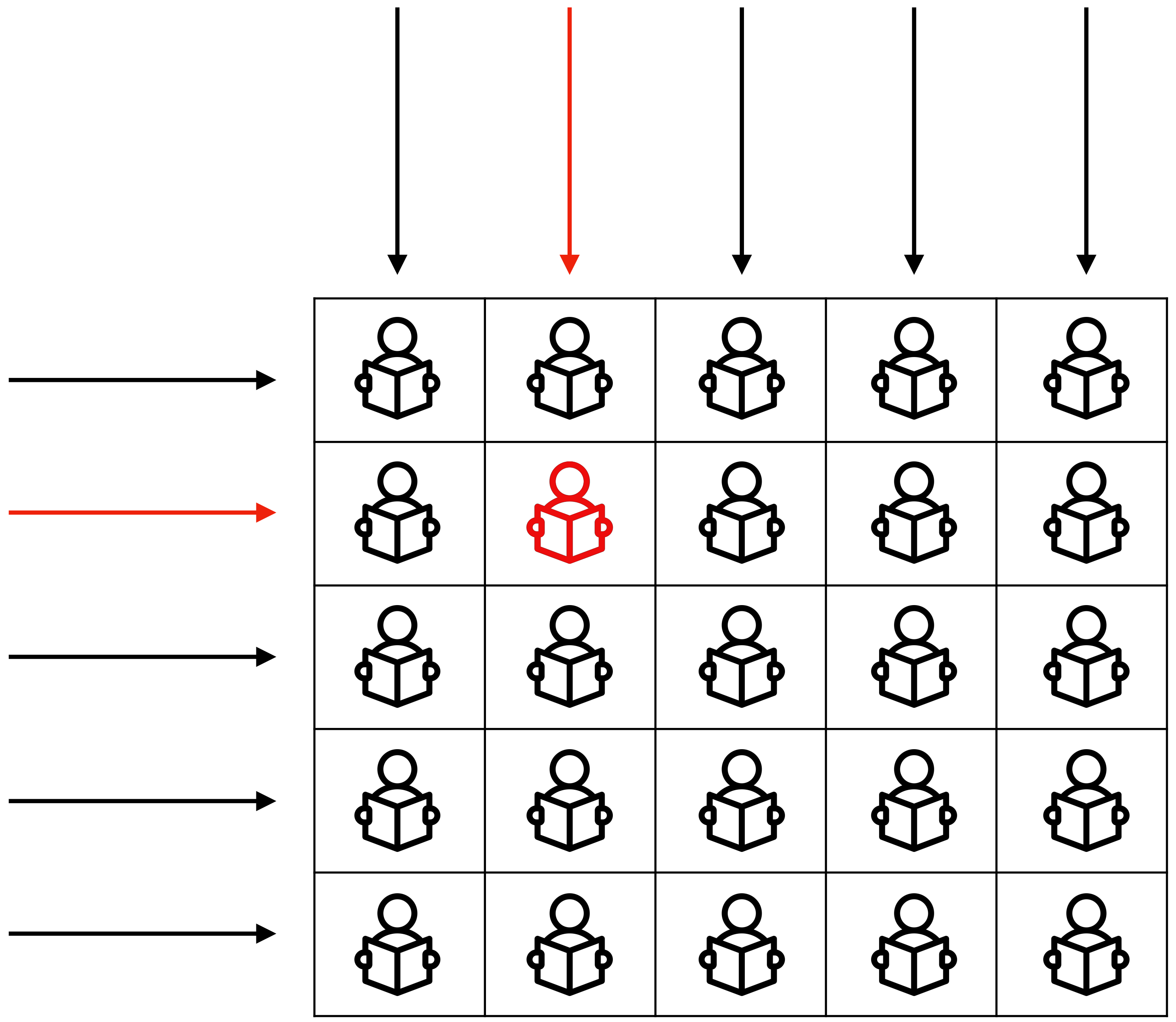
The Swiss-Army-Knife CFF



$1 - CFF(10,25)$

Some other cases

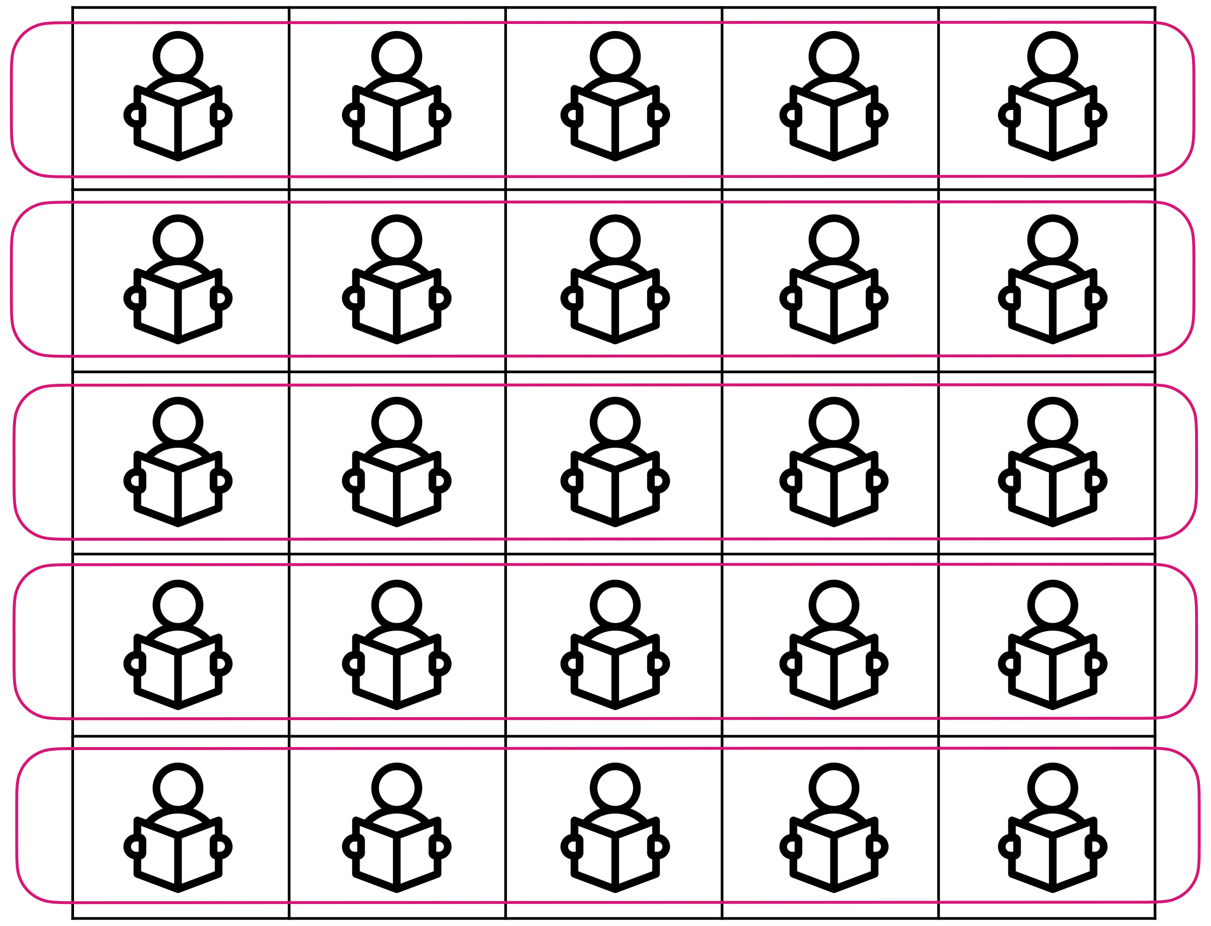
The Swiss-Army-Knife CFF



$1 - CFF(10,25)$

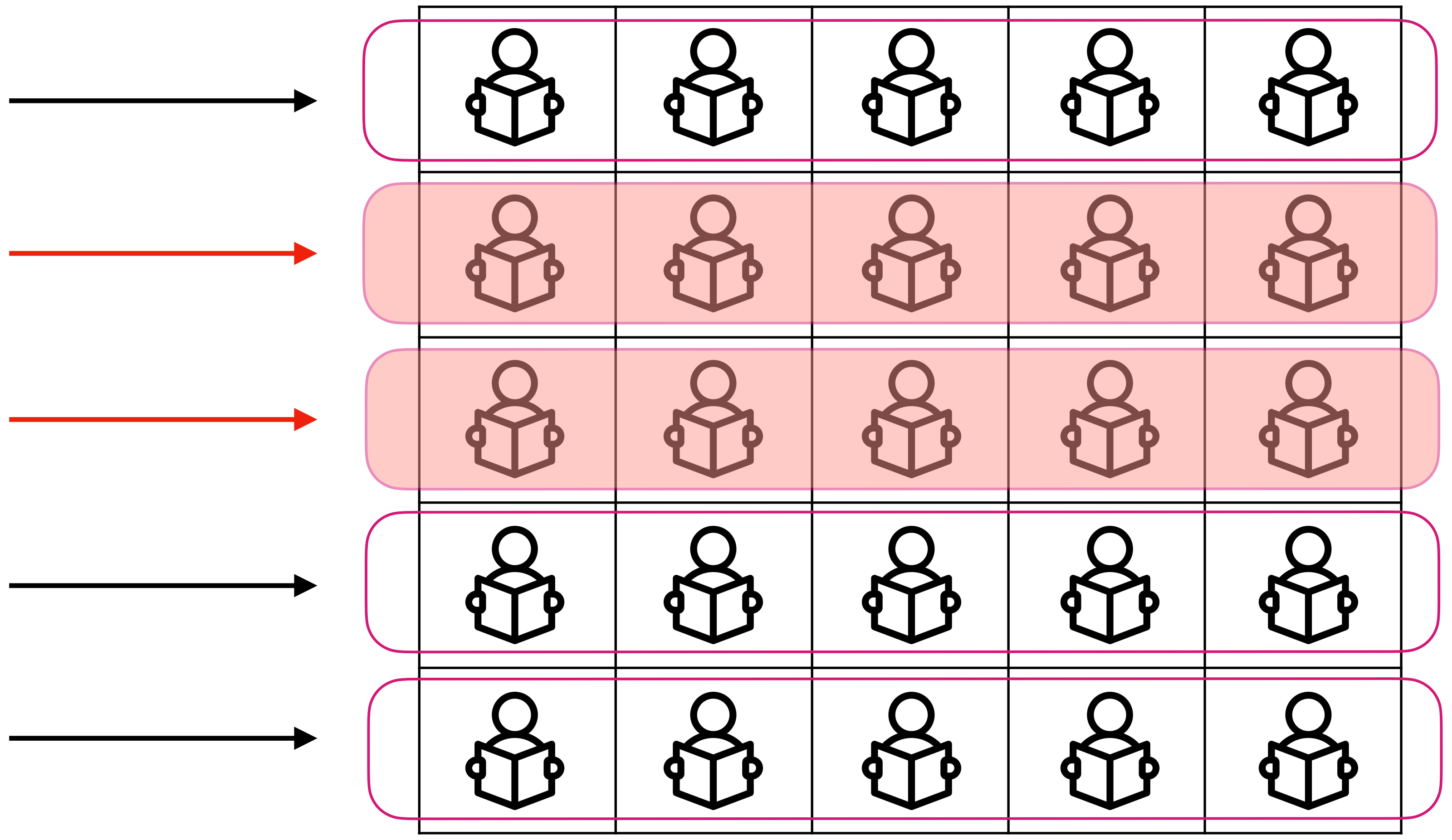
Some other cases

The Swiss-Army-Knife CFF



Some other cases

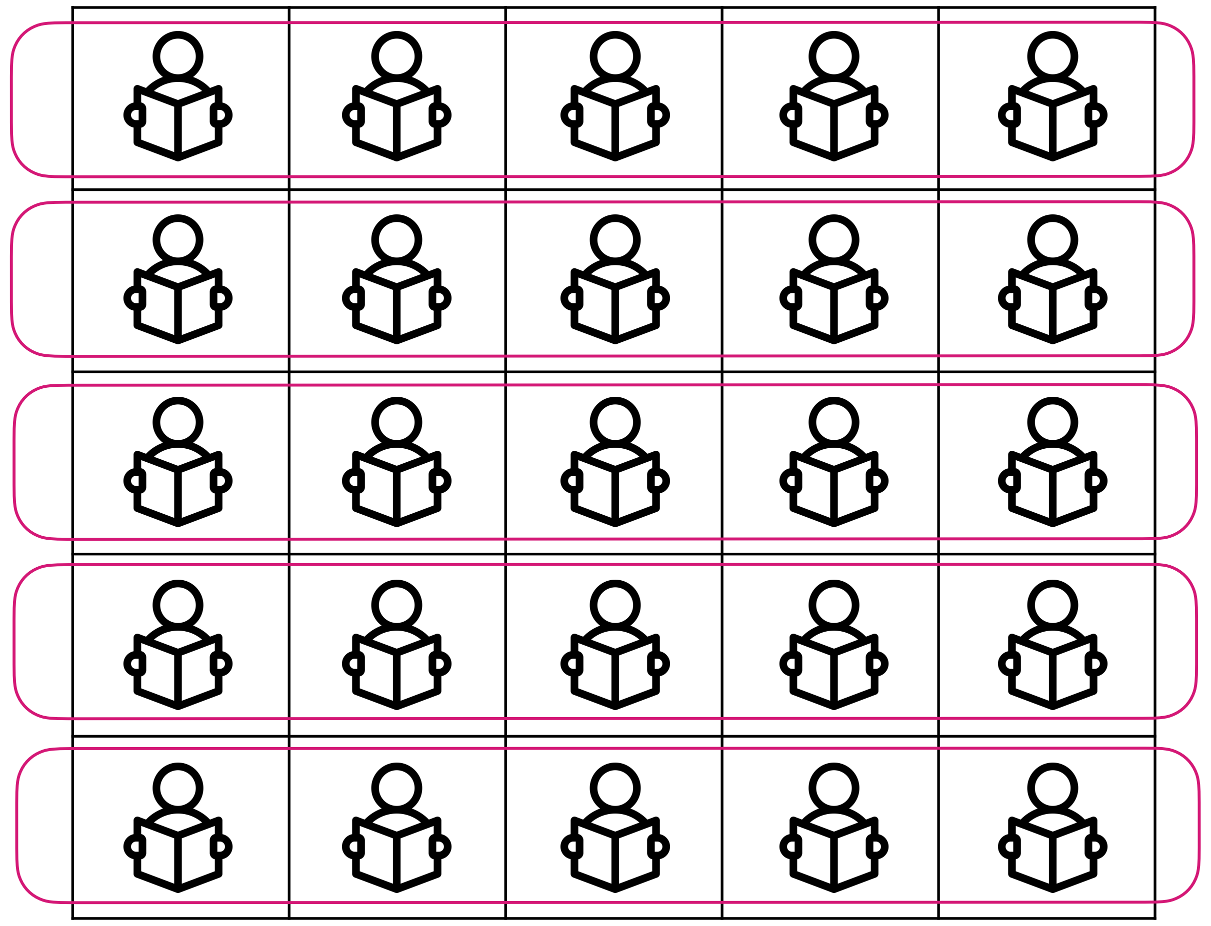
The Swiss-Army-Knife CFF



$(\mathscr{E},5) - ECFF(10,25)$

Some other cases

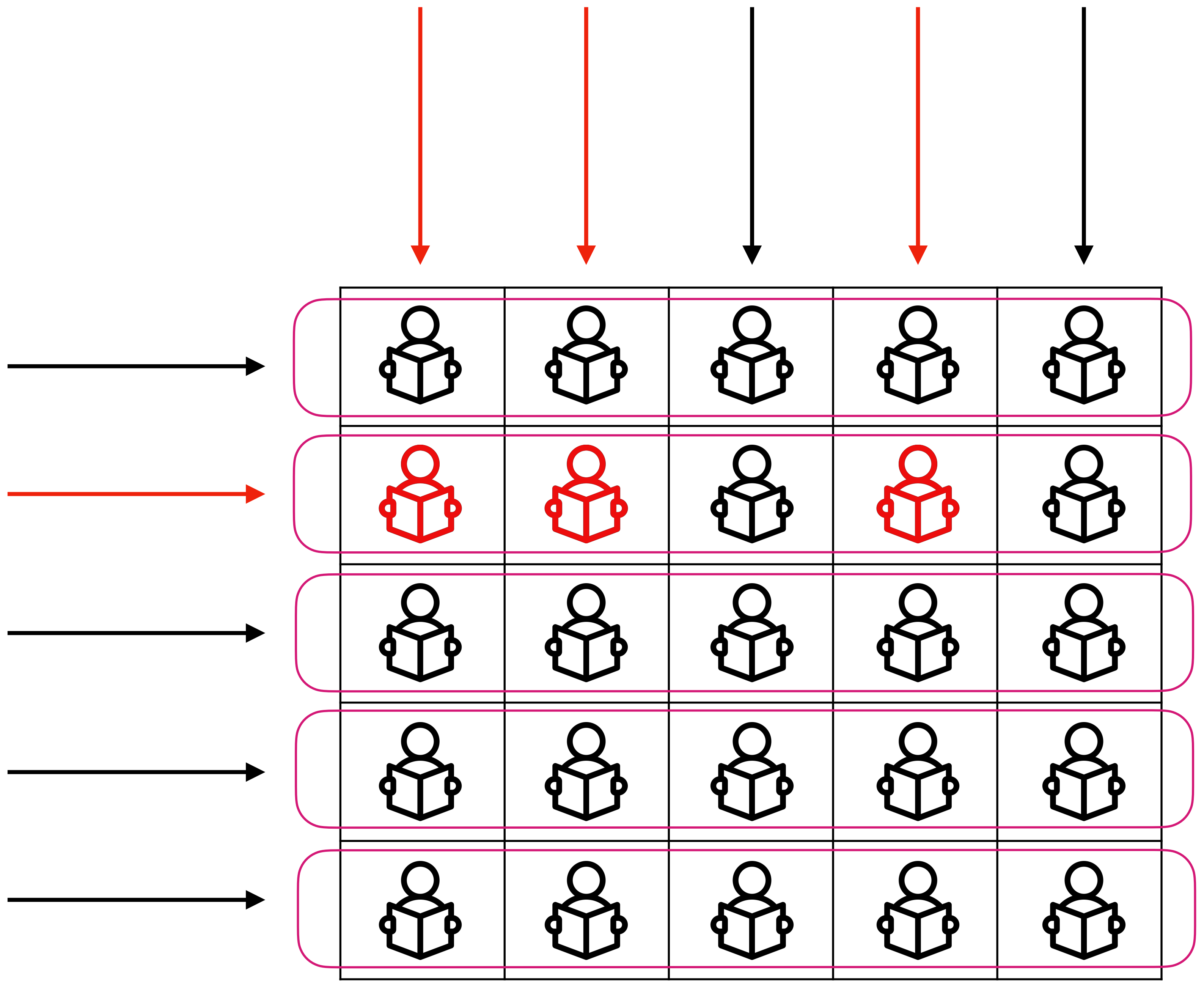
The Swiss-Army-Knife CFF



$(\mathscr{E},1) - CFF(10,25)$

Some other cases

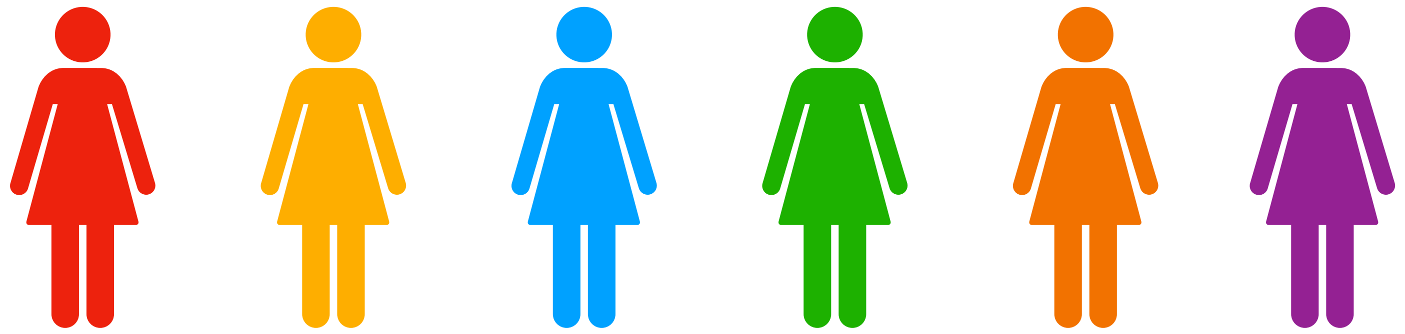
The Swiss-Army-Knife CFF



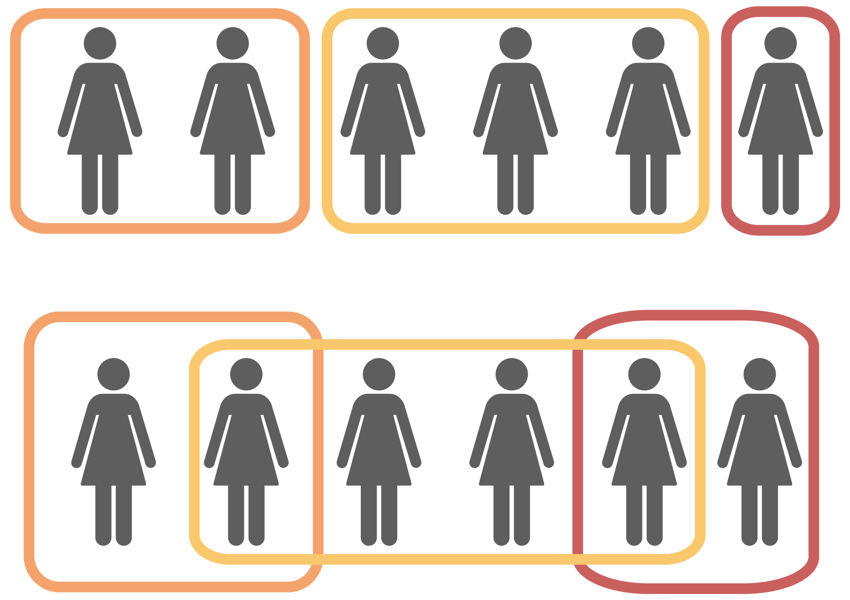
$(\mathscr{E},1) - CFF(10,25)$

In this talk

1	1	1	0	0	0
1	0	0	1	1	0
0	1	0	1	0	1
0	0	1	0	1	1



- The problem of **combinatorial group testing** in pandemic screening
- Study of ***CFFs on hypergraphs***
  - Non-overlapping edges
  - Overlapping edges
  - Some other cases
- Applications in cryptography

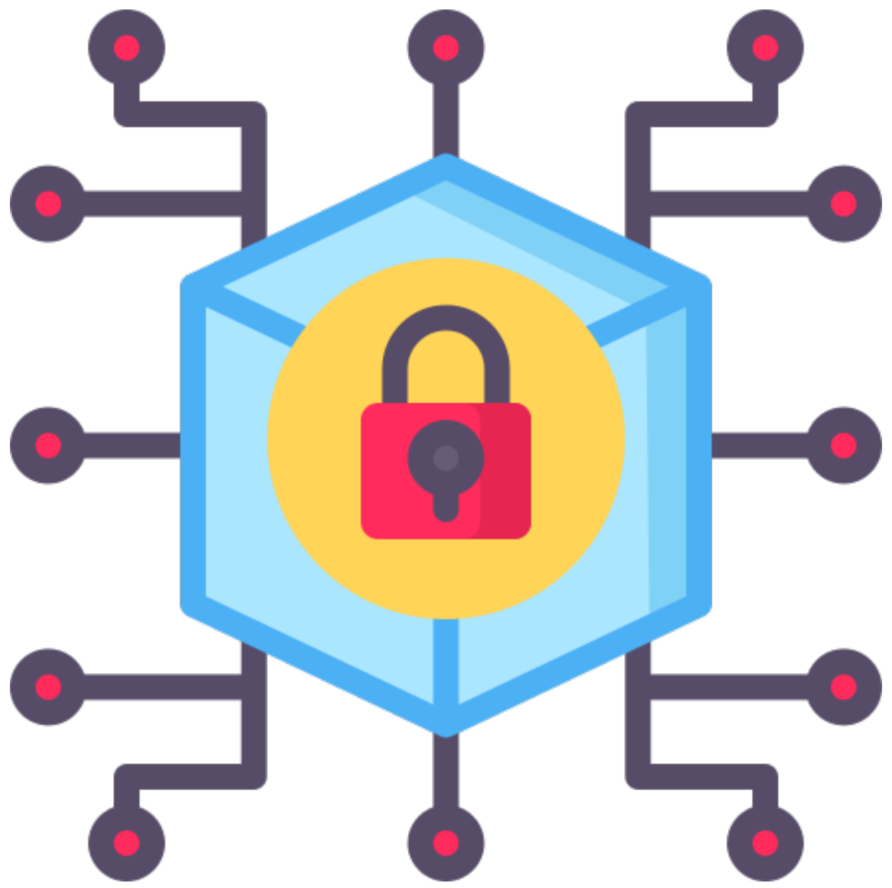


1. Bardini Idalino, T., Moura, L. Structure-Aware Combinatorial Group Testing: A New Method for PandemicScreening. In: IWOCA 2022. Lecture Notes in Computer Science, vol 13270. Springer, Cham. https://doi.org/10.1007/978-3-031-06678-8

2. Bardini Idalino, T., Moura, L. Combinatorial group testing and cover-free families on hypergraphs. 2025[Unpublished manuscript].

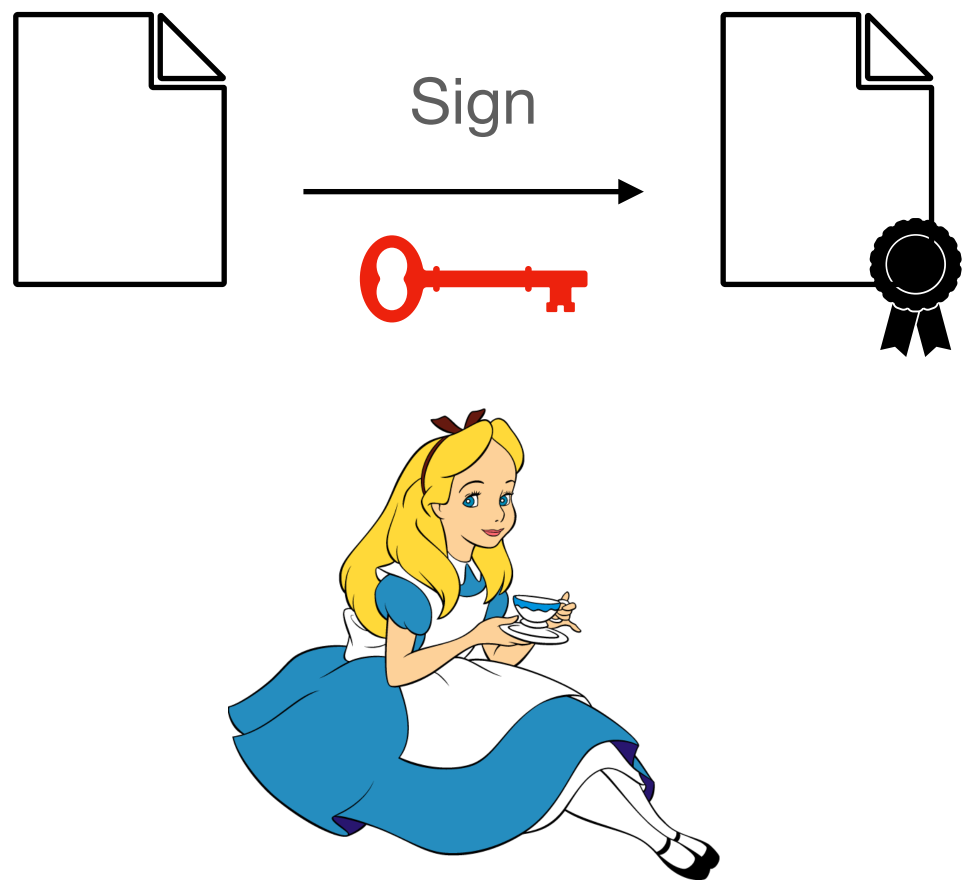
Applications in Cryptography

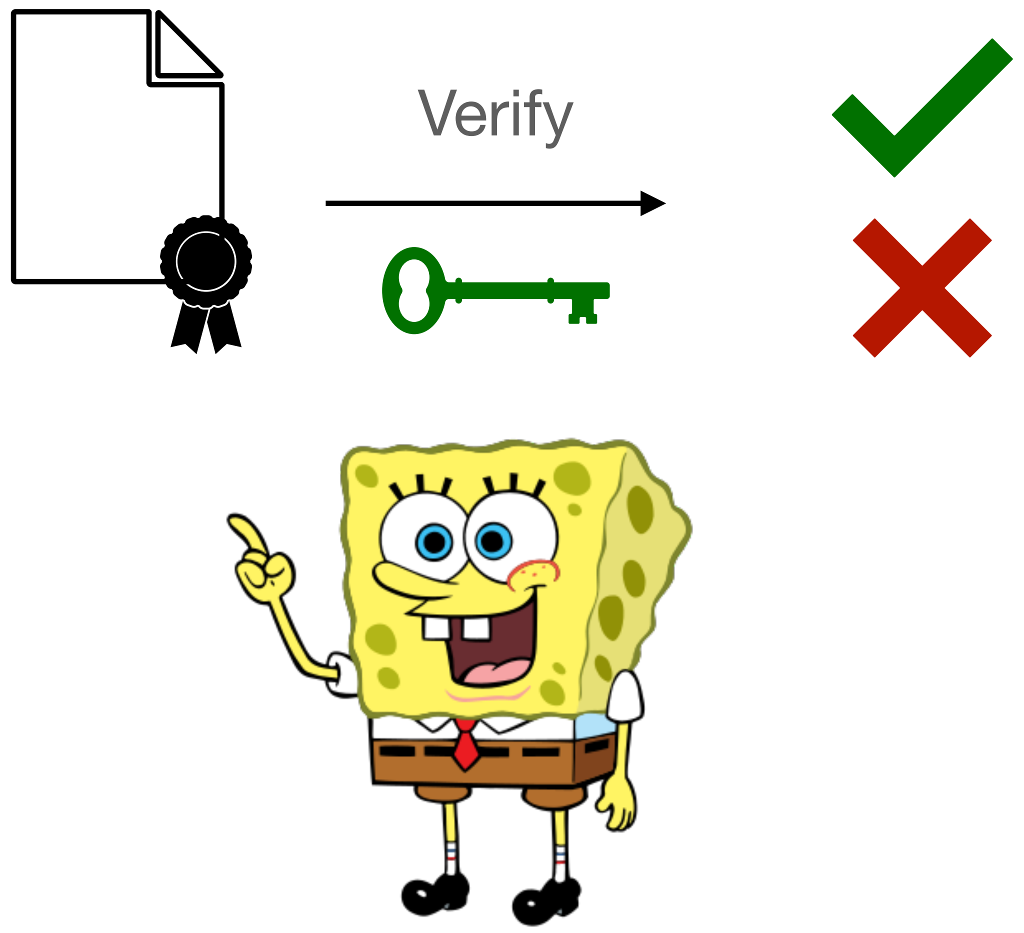
- **Fault-tolerant Digital Signatures**
  - Fault-tolerant digital signatures
    - Idalino, Moura, Custodio, Panario (2015), Idalino, Moura, Adams, (2019)
  - Fault-tolerance in aggregation of signatures
    - Zaverucha, Stinson (2010). Idalino (2015). Hartung, Kaidel, Koch, Koch, Rupp (2016). Idalino, Moura (2018, 2021)
  - Fault-tolerance in batch verification
    - Pastuszak, Pieprzyk (2000). Zaverucha, Stinson (2009).
- **Post-quantum one-time and multiple-times signature schemes**
  - Pieprzyk, Wang, Xing (2003). Zaverucha and Stinson, (2011). Kalach and Safavi-Naini (2016).
- **Key distribution**
  - Key distribution patterns
    - Mitchell and Piper (1988)
  - Broadcast authentication
    - Safavi-Naini and Wang (1998) . Ling, Wang, Xing (2007).
  - Broadcast encryption
    - Gafni, Staddon, Yin (1999). D'Arco and Stinson (2003)
  - Traitor Tracing
    - Stinson and Wei (1998). Tonien and Safavi-Naini (2006)
- and many many others..



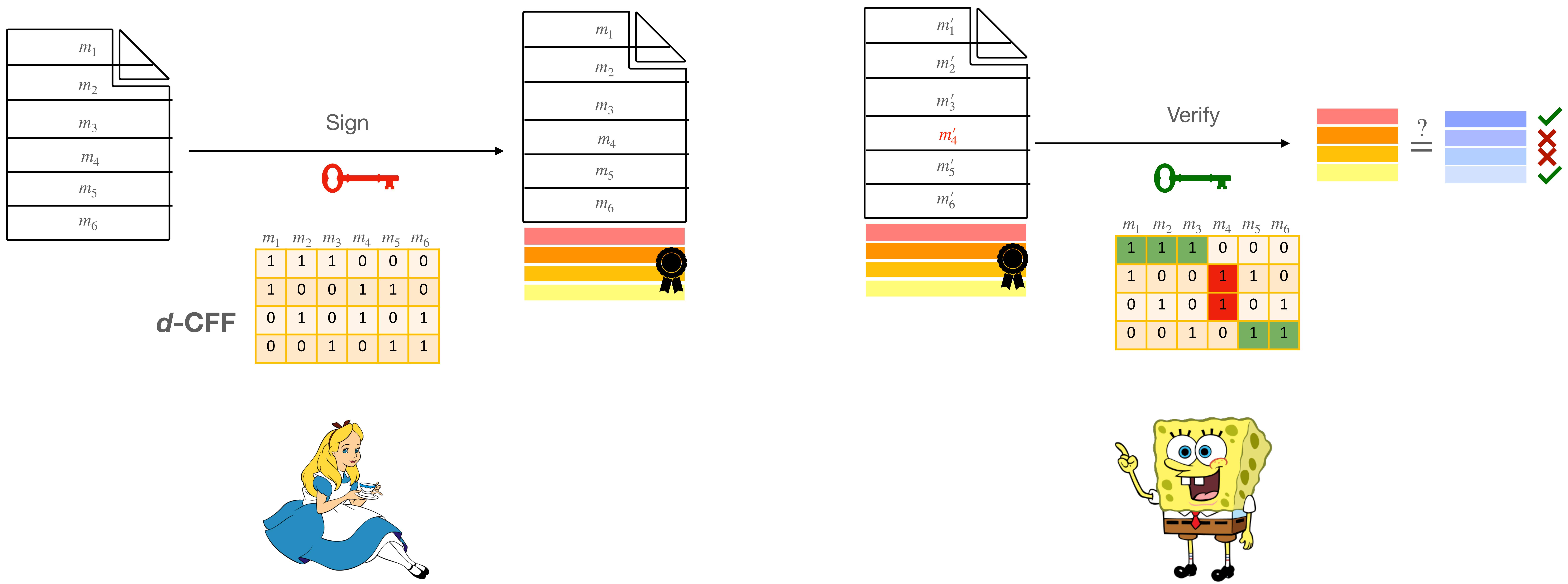
More details:
IDALINO, T. B.; MOURA, L., A Survey of Cover-Free Families: Constructions, Applications, and Generalizations. New Advances in Designs, Codes and Cryptography. 86 (2024),

Applications in cryptography





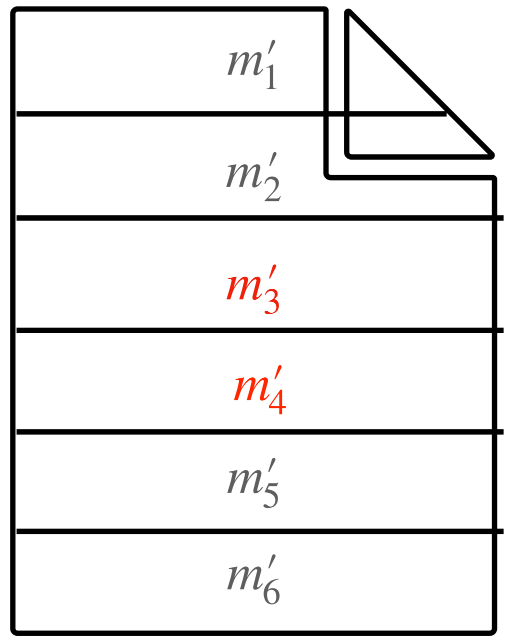
Applications in cryptography



m_1	m_2	m_3	m_4	m_5	m_6
1	1	1	0	0	0
1	0	0	1	1	0
0	1	0	1	0	1
0	0	1	0	1	1

m_1	m_2	m_3	m_4	m_5	m_6
1	1	1	0	0	0
1	0	0	1	1	0
0	1	0	1	0	1
0	0	1	0	1	1

Applications in cryptography

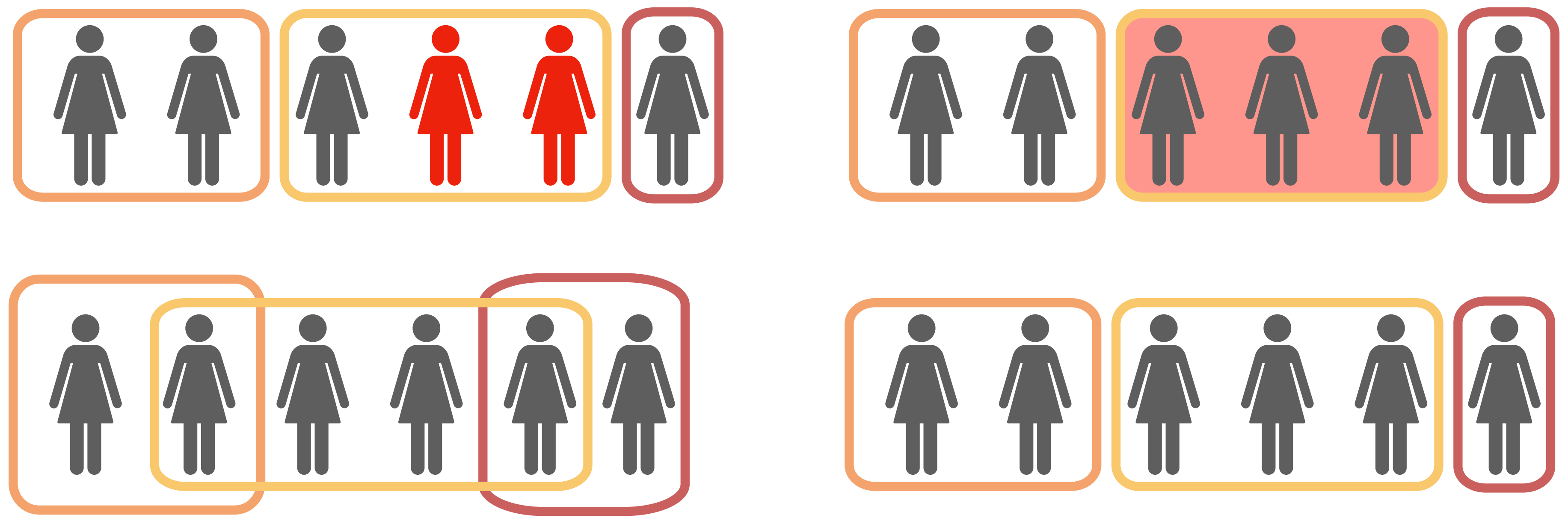


- Once modifications happened, it is natural to expect that they will happen in clusters of blocks
- This application was studied under the name of *variable CFFs* *
- It is natural to model these clusters of modifications using hypergraphs

* Idalino, T.B.: Fault tolerance in cryptographic applications using cover-free families. Ph.D. thesis, University of Ottawa, Canada (2019)

In this talk

- The problem of **combinatorial group testing** in pandemic screening
- Naturally model close contacts using hypergraphs
- Construction of *CFFs on hypergraphs*
    - Vertex-identifying and edge-identifying CFFs
    - Overlapping and non-overlapping edges



1. Bardini Idalino, T., Moura, L. Structure-Aware Combinatorial Group Testing: A New Method for PandemicScreening. In: IWOCA 2022. Lecture Notes in Computer Science, vol 13270. Springer, Cham. https://doi.org/10.1007/978-3-031-06678-8
2. Bardini Idalino, T., Moura, L. Combinatorial group testing and cover-free families on hypergraphs. 2025[Unpublished manuscript].

Future work on structure-aware CFFs

- Explore other constraints of the applications
  - Limit on number of 1s per row and/or column
- Generalize definitions to allow flexible internal identification
  - Assume a bound on the number of infected items inside an edge (instead of edge size)
- Explore probabilistic constructions
- Explore practical applications in other scenarios

Combinatorial group testing on hypergraphs

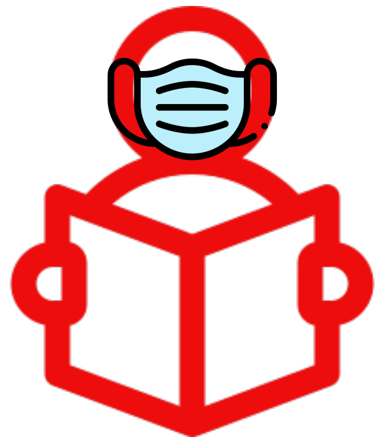
Thaís Bardini Idalino
Universidade Federal de Santa Catarina - Brazil

Ottawa Mathematics and Statistics Conference
May 2025

uOttawa

Thank you!

UFSC



Thais Bardini Idalino - thais.bardini@ufsc.br
Lucia Moura - lmoura@uottawa.ca