Tapestry: Compressed Sensing for Pooled Testing of COVID-19 Samples

Ajit Rajwade

In collaboration with:

Manoj Gopalkrishnan

Sabyasachi Ghosh, Rishi Agarwal, Mohammad Ali Rehan, Shreya Pathak, Yash Gupta, Pratyush Agarwal, Ritika Goyal, Sarthak Consul, Nimay Gupta, Prantik Chakraborty, Ritesh Goenka, Krishna Vemula, Akanksha Vyas

Interaction/collaboration with Profs. Dror Baron and Chau-Wai Wong (*NCSU*) , Chandra Murthy (*IISc*), Krishna Narayanan (*TAMU*)

Skeleton of the Talk

- Introduction and Motivation
- Computational Problem
- Noise Model
- Algorithms
- Matrix Design
- Results
- Discussion: Practical Aspects
- Discussion: Previous Work
- Summary
- More prior knowledge: contact tracing

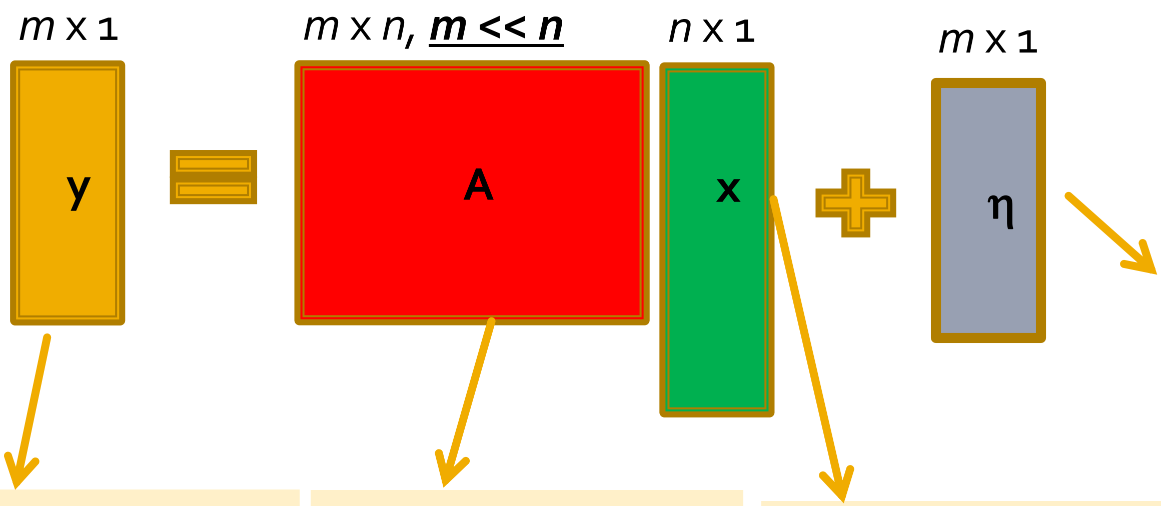
Introduction and Motivation

- The COVID-19 pandemic began in Wuhan, China.
- COVID-19 has infected more than 139 million people worldwide
- RT-PCR (*Reverse Transcription Polymerase Chain Reaction*) is the most popular method for testing a person for COVID-19
- Dearth of resources for widespread testing: time, skilled manpower, reagents, testing kits, etc.

Introduction and Motivation

- RT-PCR: naso- or oro-pharyngeal swab taken, mixed in liquid medium, tested in RT-PCR machine
- Can we **pool** (mix) subsets of *n* samples and **test the pools** to save resources?
- #pools (*m*) < #samples (*n*)
- Equal portions of participating samples are taken for creating any pool.
- A **negative pool test** implies all contributing samples are **negative (non-infected) .**
- **More work to be done if the pool tests positive (infected).**

Computational problem



Estimate **x** given **y** and **A**

Noise vector

Vector of **observed quantities** proportional to viral loads in *m* pools
m << *n*

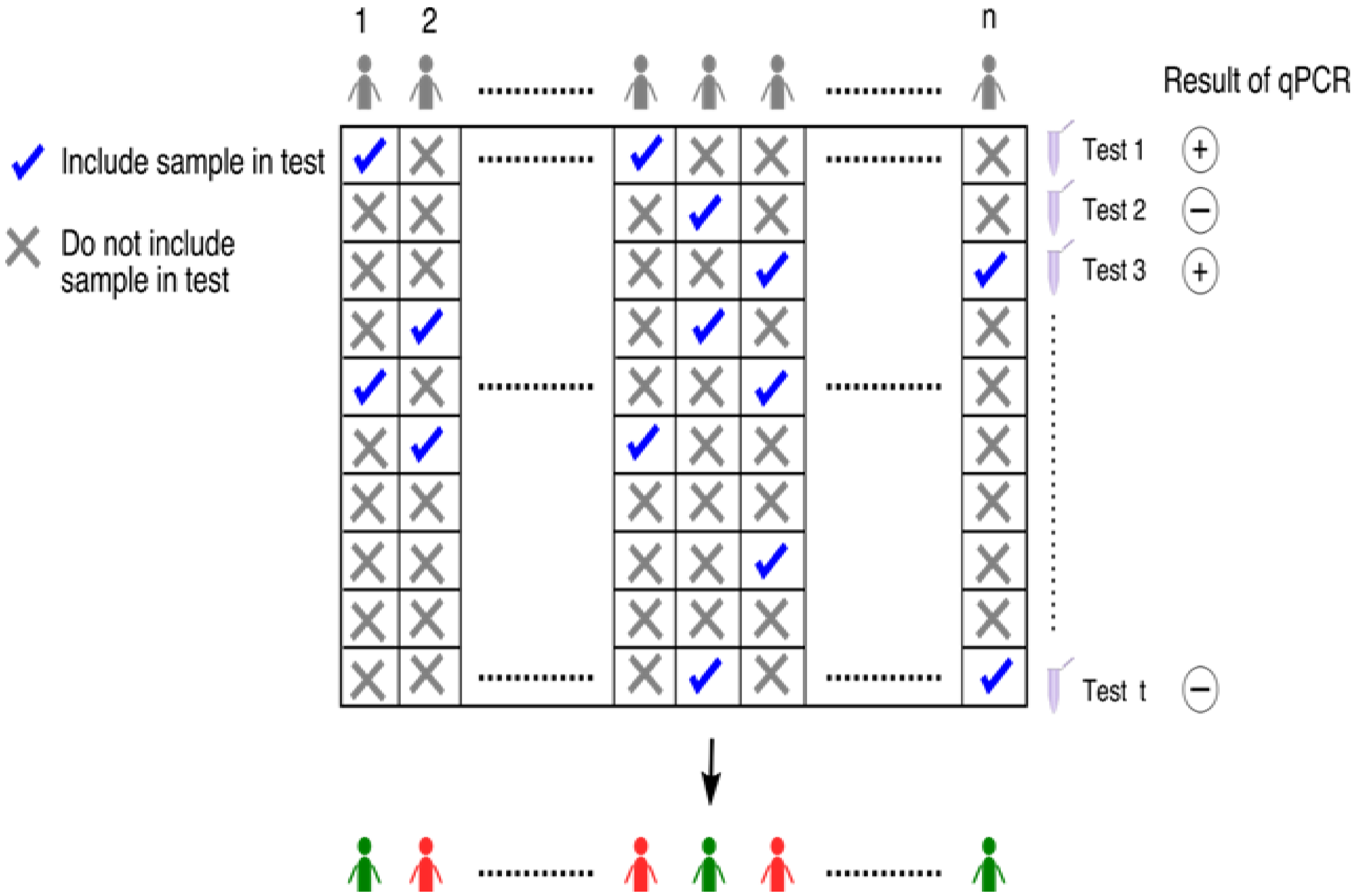
Known mixing matrix:
A_{ij} = 1 if *j*-th sample contributes to *i*-th pool, and 0 otherwise

Sparse vector of **unknown** viral loads in the samples of *n* different people

$$\min \|\boldsymbol{x}\|_1 \text{ s.t. } \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2 \leq \varepsilon.$$

$$J_{lasso}(\boldsymbol{x}; \boldsymbol{y}, \boldsymbol{A}) := \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{x}\|_1.$$

Designed pooling



1	2						n		
✓	✗	....	✓	✗	✗	....	✗	Test 1	+
✗	✗		✗	✓	✗		✗	Test 2	−
✗	✗		✗	✗	✓		✓	Test 3	+
✗	✓		✗	✓	✗		✗		
✓	✗	....	✗	✗	✓	....	✗		
✗	✓		✓	✗	✗		✗		
✗	✗		✗	✗	✗		✗		
✗	✗		✗	✗	✓		✗		
✗	✗		✗	✗	✗		✗		
✗	✗	....	✗	✓	✗	....	✓	Test t	−

RT-PCR Process

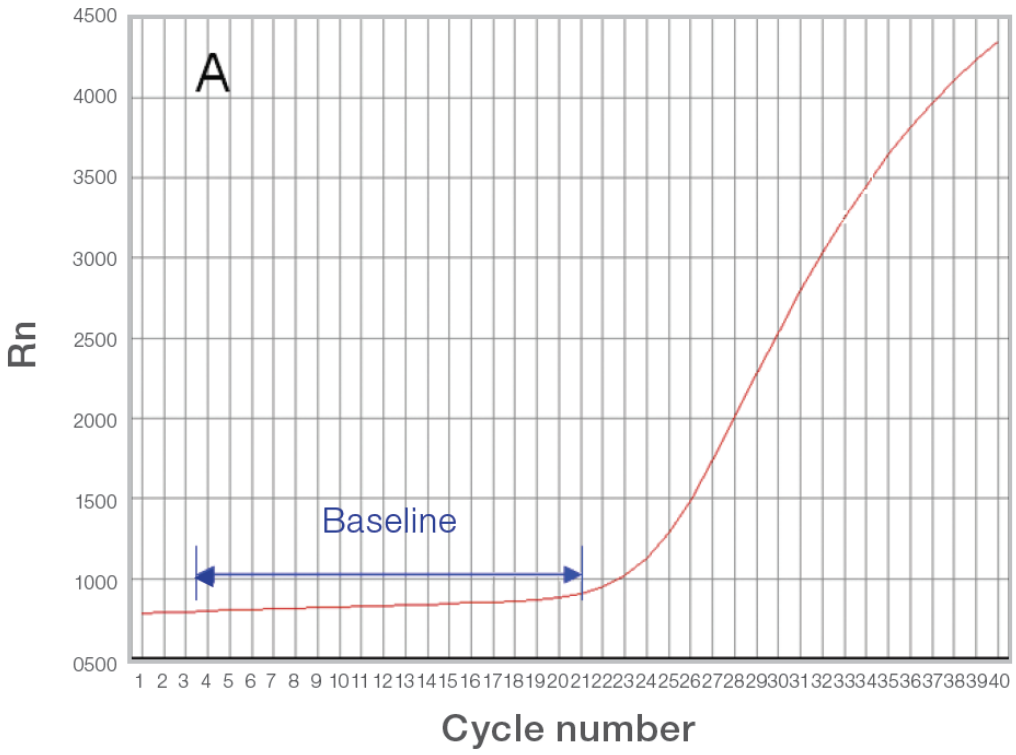
- Viral RNA molecules in the liquid medium converted to cDNA – reverse transcription
- DNA primers complementary to viral cDNA are then added
- These primers attach themselves to sections of the cDNA (if the virus is present in the sample).
- The cDNA then undergoes **exponential amplification** in the RT-PCR machine.

RT-PCR Process

- The exponential amplification occurs during cycles of alternate heating and cooling in an RT-PCR machine.

- cDNA produces **fluorescence** in response to markers – observable on a computer screen.

- The time at which fluorescence exceeds a machine-specified threshold is called the **cycle threshold** (CT) – measured by the RT-PCR machine.



Thermofisher document

RT-PCR Process

- Smaller CT value = greater viral load.
- Typical CT values range from 16 to 40; can detect even single molecules (CT ~ 40)
- Single RT-PCR test: 3-4 hours (**non-adaptive** tests are desirable)

Noise Model: RT-PCR

Initial viral count of the *i*-th pool, 1 <= *i* <= m

Noisy fluorescence threshold, and **measured** time t_i at which it was crossed

$$z_i(1+q)^{t_i} = F + \delta$$

$$\therefore \log F - t_i \log(1+q) = \log z_i + \delta / F$$

$$z_i = \boldsymbol{A}^i \boldsymbol{w}$$

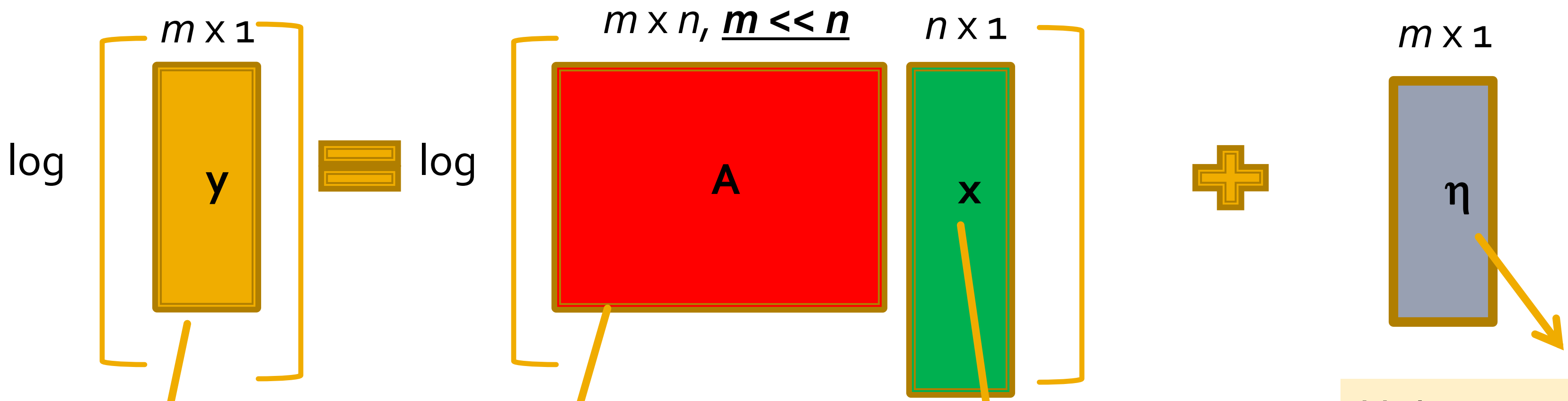
$$\boldsymbol{w} \sim \text{Poisson}(\boldsymbol{x})$$

$$\log y_i = \log \boldsymbol{A}^i \boldsymbol{x} + \eta_i \;\; ; \;\; \eta_i \sim N(0, \sigma^2)$$

Vector of *n* viral loads

Approx. in case of high viral loads

Noise model: RT-PCR



Noise vector

Vector of **observed quantities** proportional to viral loads in *m* pools
m << *n*

Known mixing matrix:
A_{ij} = 1 if *j*-th sample contributes to *i*-th pool, and 0 otherwise

Sparse vector of **unknown** viral loads in the samples of *n* different people

Estimate **x** given **y** and **A**

RT-PCR: 0-valued measurements

- If the viral load z_i in the *i*-th pool is 0, the fluorescence threshold is **never crossed** (negative test).

- In this case, all participating samples x_j (i.e. for which $A_{ij} = 1$) are **non-infected** and can be removed from further computation.

- If viral loads in infected samples are high, there is no conflation between negative and positive tests (negative result is **noiseless**).

RT-PCR: 0-valued measurements

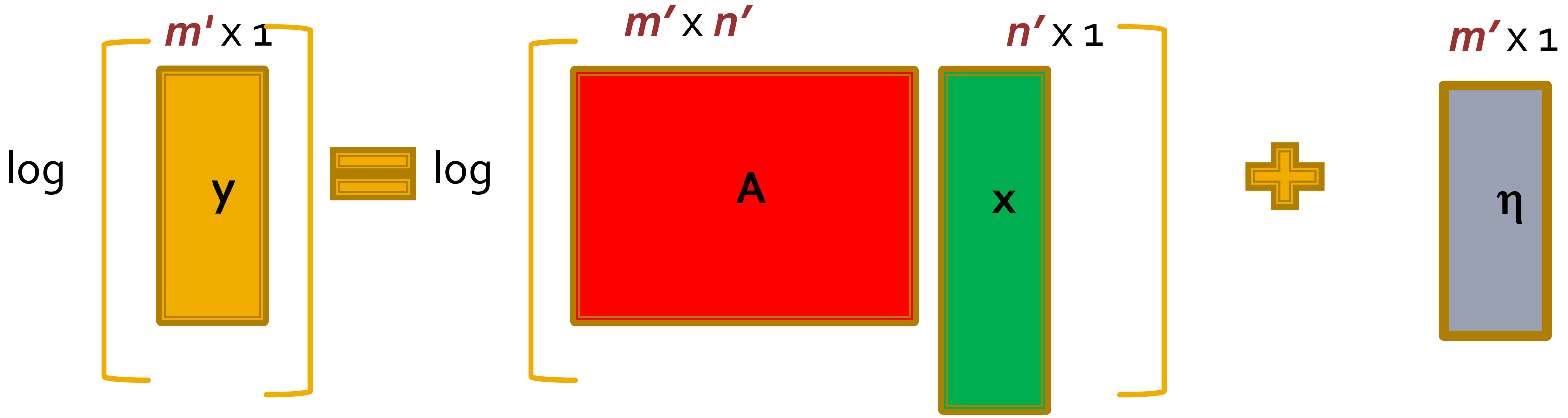
- “Removing negative samples” ≈ combinatorial group testing algorithm called **COMP** (Combinatorial Orthogonal Matching Pursuit)

- But COMP regards the other samples to be positive (possible **false positives**)

- After discarding negative samples, we may also get a few **sure positives** – positive tests with only one non-discarded contributing sample.

Compressed Sensing Algorithms

- We now have a **smaller-sized** **y**, **A**, **x** after discarding the negatives in **x**, and pruning **y**, **A** accordingly.



Compressed Sensing Algorithms

- Run after the “COMP-step”
- Non-negative LASSO

$$\arg\min_{\boldsymbol{x}} \| \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x} \|^2 + \lambda \|\boldsymbol{x}\|_1 \text{ s.t. } \boldsymbol{x} >= \boldsymbol{0}$$

- Non-negative Orthogonal Matching Pursuit
- **Sparse Bayesian Learning** (E-M algorithm)

$$p(x_i; \varphi_i) = \frac{\exp(-x_i^2/(2\varphi_i))}{\sqrt{2\pi\varphi_i}}; \varphi_i \geq 0.$$

- Brute-force search (for small *k* only)

Mixing matrix design

- Many methods exist:

✓ (*) Minimize mutual coherence of **A**

✓ (#) Maximize mutual information between **Ax** and **x** [*requires estimating high-D PDFs from representative signals*]

✓ ($) Minimize MMSE assuming Gaussian noise in **y** and GMM for **x** [*requires GMM fitting on representative signals*].

(*) Abdolghosemi et al, "On the optimization of the measurement matrix for CS"
(#) L Wang et al, "Information-theoretic measurement matrix design"
($) Renna et al, "Reconstruction of Signals Drawn from a Gaussian Mixture via Noisy Compressive Measurements"

Mixing matrix design

- Constraints on **A** (size *m* x *n*):

✓ Should allow for unique recovery of sparse **x** from **A** and noisy **y** : no sparse vector except **o** should lie in *nullspace*(**A**)

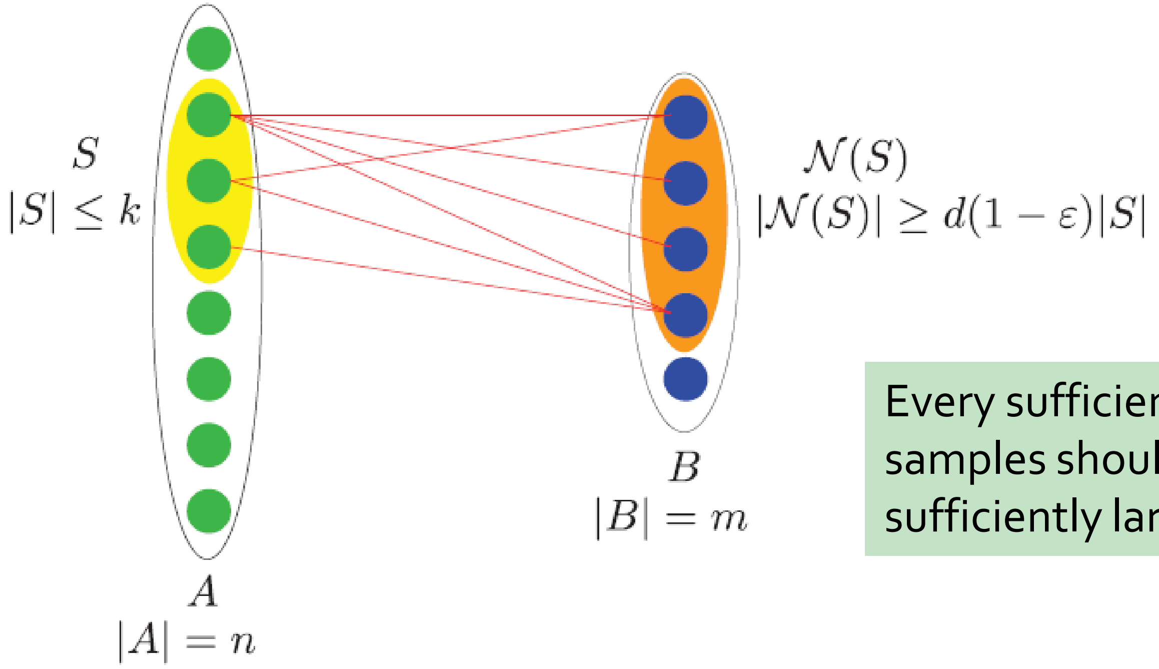
✓ Non-negative

✓ **Binary** and **sparse** (for ease of sample pooling in the lab)

✓ Should be **flexible**, i.e. should adapt easily to slightly different sizes

Mixing matrix design: Expanders

- **Expander matrices:** obey RIP-1, i.e. preserve norm of k-sparse vectors



Every sufficiently small subset of samples should participate in a sufficiently large number of pools

Fig. 1. A (k, ϵ)-expander. In this example, the green nodes correspond to A, the blue nodes correspond to B, the yellow oval corresponds to the set $S \subset A$, and the orange oval corresponds to the set $\mathcal{N}(S) \subset B$. There are three colliding edges.

Mixing matrix design: Expanders

- A randomly generated left-regular graph with left-degree *d* is an expander – with high probability.

- Can be converted into a graph/matrix that is left-regular as well as right-regular with almost same parameters.

- But difficult to verify whether a **given** matrix is an expander.

Mixing matrix design: deterministic matrices

- Many methods for **deterministic** sensing matrix design (*)
- We use **Steiner triple matrices** of size *m* x *n* where *n* = C(*m*,2)/3
  - Each column has at most 3 ones (out of *m* elements)
  - No two columns have a dot-product more than 1 (good **mutual coherence**)

(*) Devore, Deterministic constructions of compressed sensing matrices, 2007

(*) Naidu et al, Deterministic CS matrices: construction via Euler squares, 2016

Mixing matrix design: deterministic matrices

- In particular, we use **Kirkman triples** – a type of Steiner triples where each block of consecutive *m*/3 columns (starting from index 0) sum up to **1**.

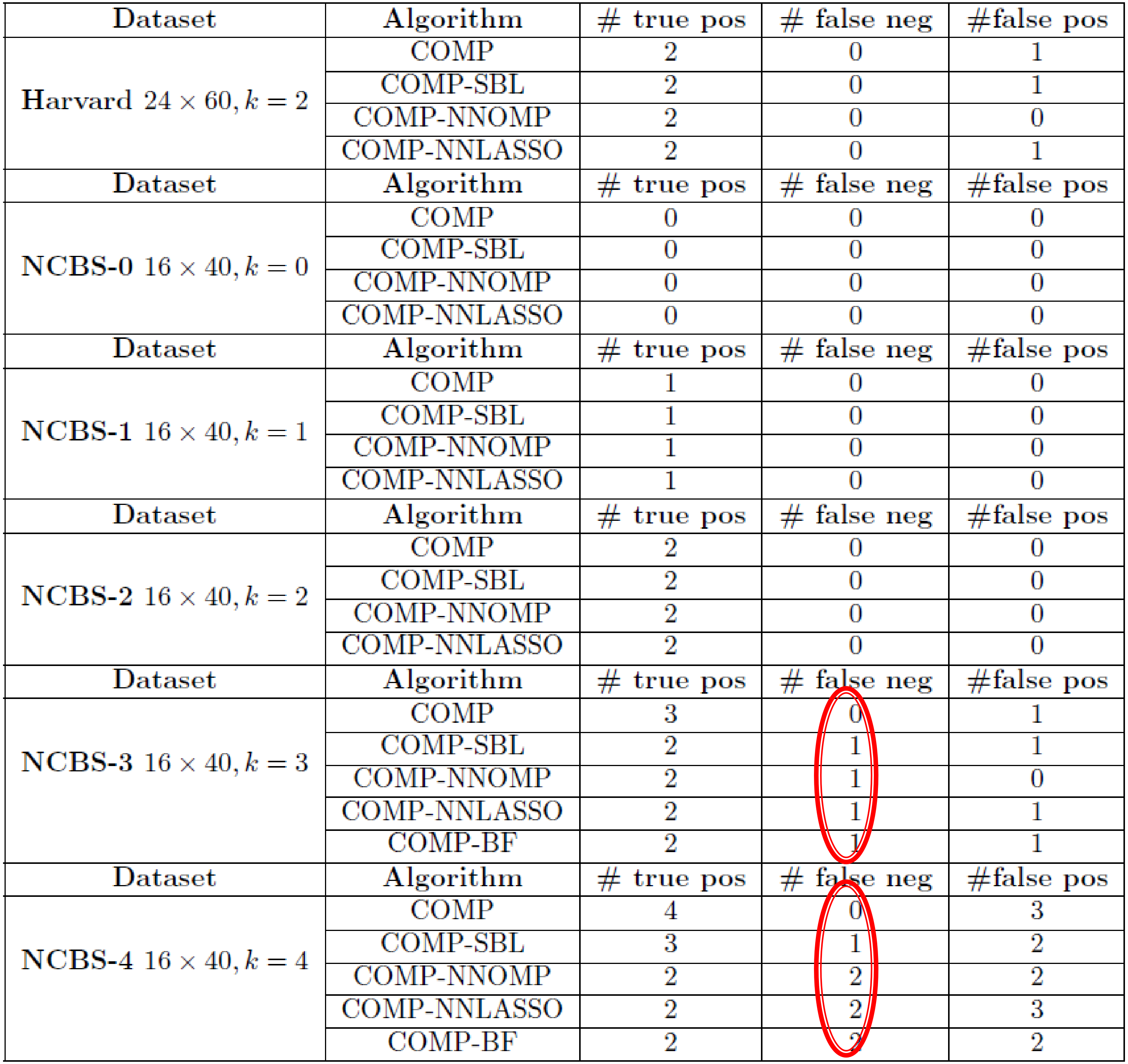
- Kirkman triple matrices have flexible sizes: *n* can be any integer multiple of *m*/3 and still each pool has the same number of samples.

Results

- Synthetic/simulated data – refer our arxiv paper
- Real data from labs at NCBS and Harvard, obtained by artificially injecting viral RNA copies in samples
- Pools created using our designed mixing matrix **A**.
- Results on **blinded** experiments (no idea of ground truth before reporting of results, or during creation of pools by technicians)

Results: figures of Merit

- RMSE between estimated and true viral load
- Sensitivity = #correctly detected positives/#true positives
- Specificity = #correctly detected negatives/#true negatives



Dataset	Algorithm	# true pos	# false neg	#false pos
Harvard $24 \times 60, k = 2$	COMP	2	0	1
	COMP-SBL	2	0	1
	COMP-NNOMP	2	0	0
	COMP-NNLASSO	2	0	1
Dataset	Algorithm	# true pos	# false neg	#false pos
NCBS-0 $16 \times 40, k = 0$	COMP	0	0	0
	COMP-SBL	0	0	0
	COMP-NNOMP	0	0	0
	COMP-NNLASSO	0	0	0
Dataset	Algorithm	# true pos	# false neg	#false pos
NCBS-1 $16 \times 40, k = 1$	COMP	1	0	0
	COMP-SBL	1	0	0
	COMP-NNOMP	1	0	0
	COMP-NNLASSO	1	0	0
Dataset	Algorithm	# true pos	# false neg	#false pos
NCBS-2 $16 \times 40, k = 2$	COMP	2	0	0
	COMP-SBL	2	0	0
	COMP-NNOMP	2	0	0
	COMP-NNLASSO	2	0	0
Dataset	Algorithm	# true pos	# false neg	#false pos
NCBS-3 $16 \times 40, k = 3$	COMP	3	0	1
	COMP-SBL	2	1	1
	COMP-NNOMP	2	1	0
	COMP-NNLASSO	2	1	1
	COMP-BF	2	1	1
Dataset	Algorithm	# true pos	# false neg	#false pos
NCBS-4 $16 \times 40, k = 4$	COMP	4	0	3
	COMP-SBL	3	1	2
	COMP-NNOMP	2	2	2
	COMP-NNLASSO	2	2	3
	COMP-BF	2	2	2

k = # of infected samples out of *n*

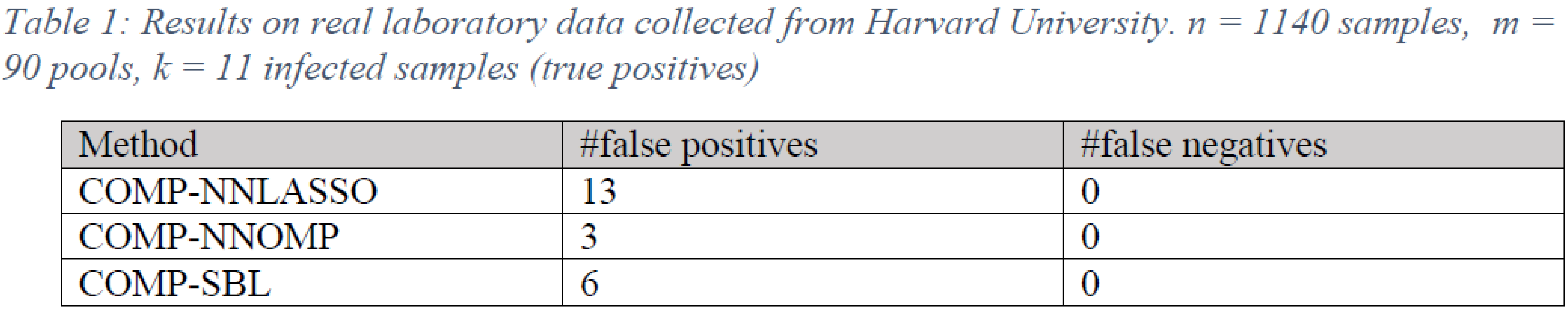


Table 1: Results on real laboratory data collected from Harvard University. $n = 1140$ samples, $m = 90$ pools, $k = 11$ infected samples (true positives)

Method	#false positives	#false negatives
COMP-NNLASSO	13	0
COMP-NNOMP	3	0
COMP-SBL	6	0

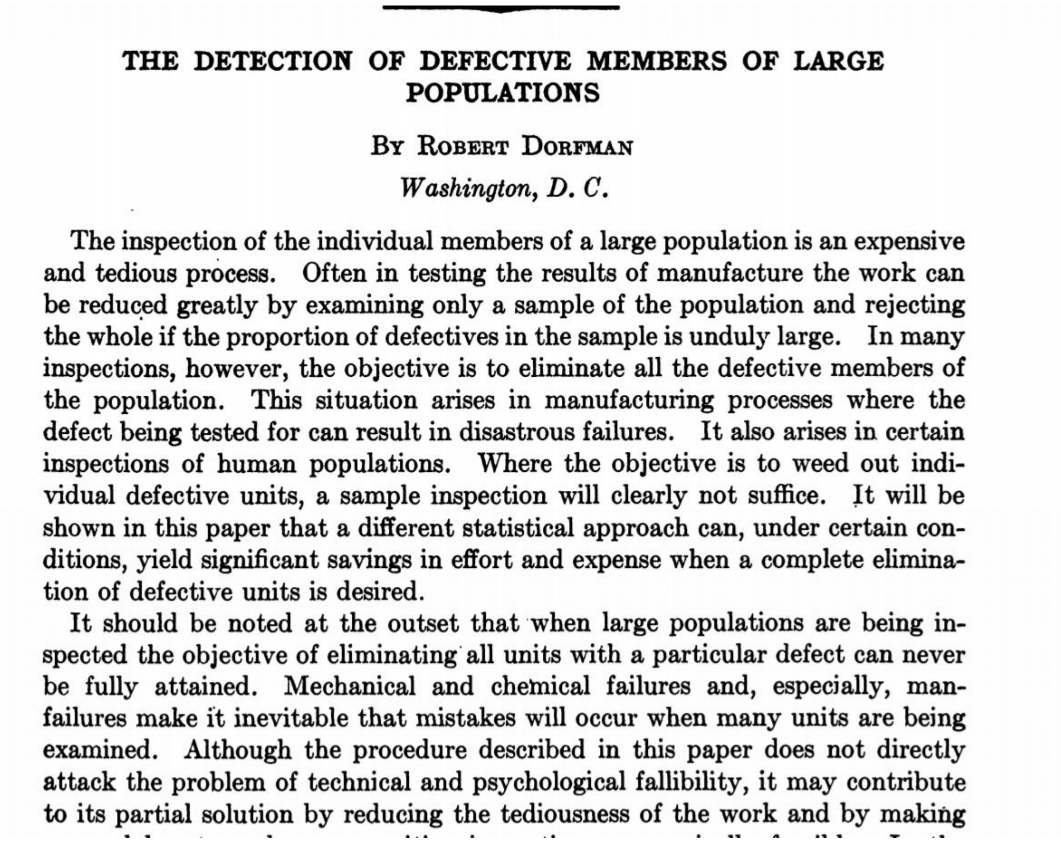
Experiment in a lab at Harvard, using a liquid handling robot

Clinical Trials with COVID19 samples

- ✓ 72 samples in 36 tests with upto 12 positives
- ✓ 105 samples in 45 tests with upto 15 positives
- ✓ 320 samples in 48 tests with 5 positives
- ✓ 500 samples in 60 tests with 10 positives
- ✓ 961 samples in 93 tests with 10 positives

✓ Very small number of false positives, no false negatives

A word about Dorfman Pooling



THE DETECTION OF DEFECTIVE MEMBERS OF LARGE POPULATIONS

By Robert Dorfman

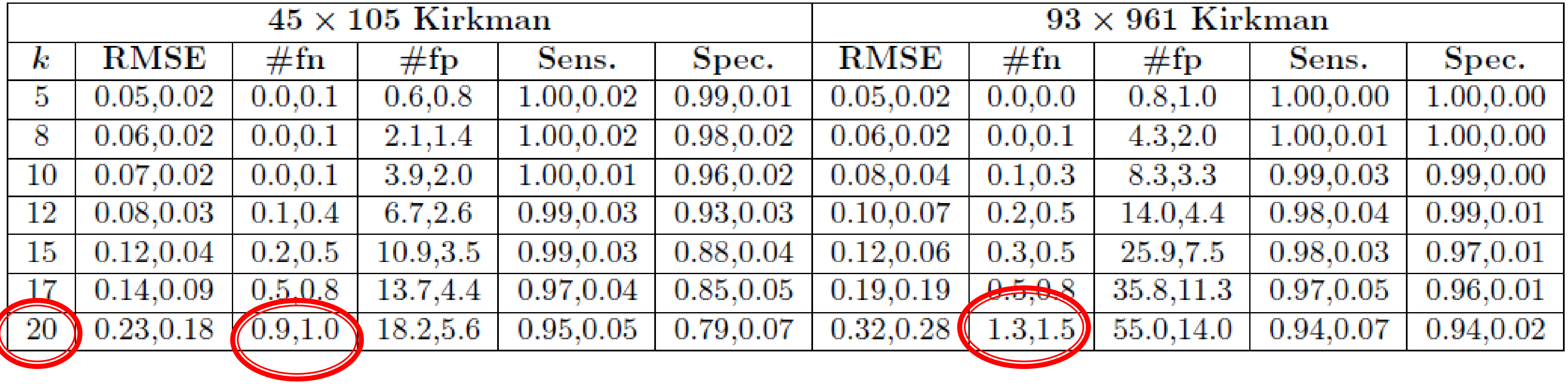
Washington, D. C.

The inspection of the individual members of a large population is an expensive and tedious process. Often in testing the results of manufacture the work can be reduced greatly by examining only a sample of the population and rejecting the whole if the proportion of defectives in the sample is unduly large. In many inspections, however, the objective is to eliminate all the defective members of the population. This situation arises in manufacturing processes where the defect being tested for can result in disastrous failures. It also arises in certain inspections of human populations. Where the objective is to weed out individual defective units, a sample inspection will clearly not suffice. It will be shown in this paper that a different statistical approach can, under certain conditions, yield significant savings in effort and expense when a complete elimination of defective units is desired.

It should be noted at the outset that when large populations are being inspected the objective of eliminating all units with a particular defect can never be fully attained. Mechanical and chemical failures and, especially, man-failures make it inevitable that mistakes will occur when many units are being examined. Although the procedure described in this paper does not directly attack the problem of technical and psychological fallibility, it may contribute to its partial solution by reducing the tediousness of the work and by making

- Testing in two stages (one after the other)
- Adaptive method
- First round: create $(n/k)^{0.5}$ pools of size $(nk)^{0.5}$ each; k = #infected people
- Second round: individual testing of each person in a positive pool
- Two of rounds of RT-PCR = time consuming

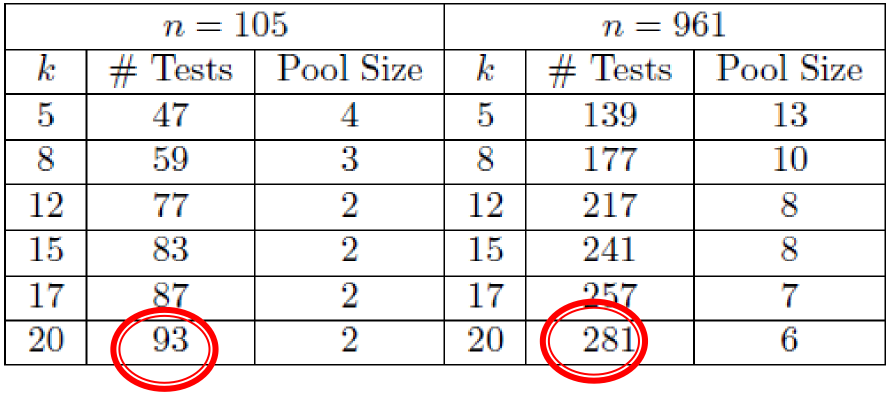
Comparing COMP-SBL (one-stage) with two-stage Dorfman pooling



	45 × 105 Kirkman					93 × 961 Kirkman				
k	RMSE	#fn	#fp	Sens.	Spec.	RMSE	#fn	#fp	Sens.	Spec.
5	0.05,0.02	0.0,0.1	0.6,0.8	1.00,0.02	0.99,0.01	0.05,0.02	0.0,0.0	0.8,1.0	1.00,0.00	1.00,0.00
8	0.06,0.02	0.0,0.1	2.1,1.4	1.00,0.02	0.98,0.02	0.06,0.02	0.0,0.1	4.3,2.0	1.00,0.01	1.00,0.00
10	0.07,0.02	0.0,0.1	3.9,2.0	1.00,0.01	0.96,0.02	0.08,0.04	0.1,0.3	8.3,3.3	0.99,0.03	0.99,0.00
12	0.08,0.03	0.1,0.4	6.7,2.6	0.99,0.03	0.93,0.03	0.10,0.07	0.2,0.5	14.0,4.4	0.98,0.04	0.99,0.01
15	0.12,0.04	0.2,0.5	10.9,3.5	0.99,0.03	0.88,0.04	0.12,0.06	0.3,0.5	25.9,7.5	0.98,0.03	0.97,0.01
17	0.14,0.09	0.5,0.8	13.7,4.4	0.97,0.04	0.85,0.05	0.19,0.19	0.5,0.8	35.8,11.3	0.97,0.05	0.96,0.01
20	0.23,0.18	0.9,1.0	18.2,5.6	0.95,0.05	0.79,0.07	0.32,0.28	1.3,1.5	55.0,14.0	0.94,0.07	0.94,0.02

Table 3: Performance of COMP followed by SBL (on synthetic data) for 45 × 105 and 93 × 961 Kirkman triple matrices. For each criterion, mean and standard deviation values are reported, across 100 signals.

k = # of infected samples out of *n*

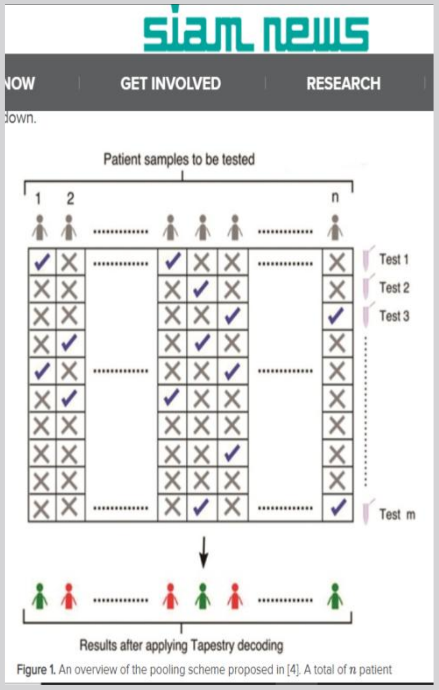


	$n = 105$			$n = 961$	
k	# Tests	Pool Size	k	# Tests	Pool Size
5	47	4	5	139	13
8	59	3	8	177	10
12	77	2	12	217	8
15	83	2	15	241	8
17	87	2	17	257	7
20	93	2	20	281	6

Table 6: Number of tests needed by optimal Dorfman Testing for number of samples (n) 105 and 961 for various k. Note that the compressed sensing methods we have adopted here require much fewer tests (45 and 93) typically, and do not require two rounds of testing.

Outputs

- We have built a system called Tapestry.
- Phone App called Byom.
- Papers on medarxiv and arxiv.
- Work cited in newsletters of IEEE Signal Processing Society and SIAM.
- David Donoho’s talk at SIAM



siam news

NOW GET INVOLVED RESEARCH

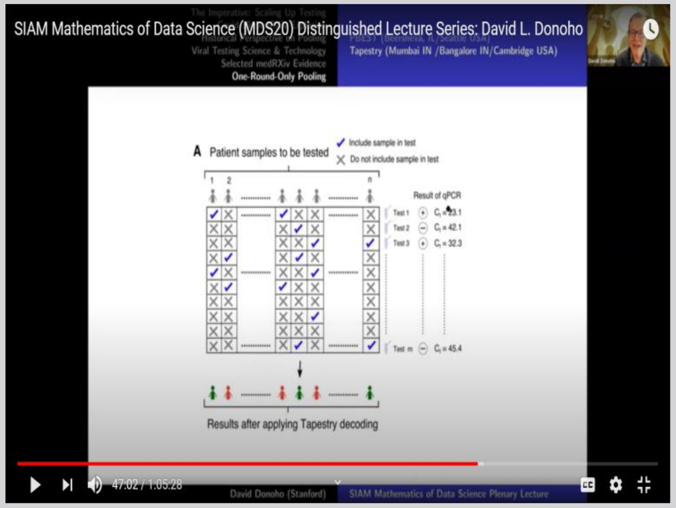
down.

Patient samples to be tested
1
2
n
Test 1
Test 2
Test 3
Test m
Results after applying Tapestry decoding

Figure 1. An overview of the pooling scheme proposed in [4]. A total of n patient

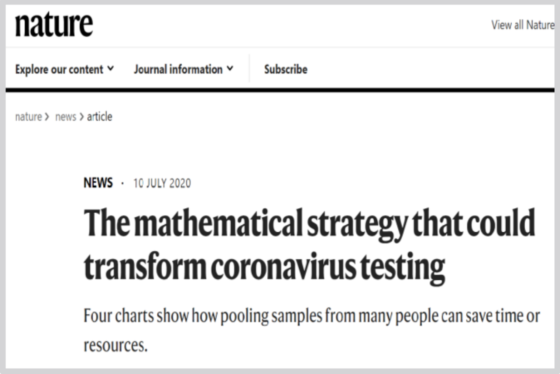
Soumya Swaminathan @doctorsoumya

The mathematical strategy that could transform coronavirus testing



SIAM Mathematics of Data Science (MDS20) Distinguished Lecture Series: David L. Donoho

The Imperative: Scaling Up Testing
Historical Perspective on Pooling
Viral Testing Science & Technology
Selected medRXiv Evidence
One-Round-Only Pooling
PBEST (Beersheva, IL/Seattle USA)
Tapestry (Mumbai IN /Bangalore IN/Cambridge USA)
A Patient samples to be tested
✓ Include sample in test
✕ Do not include sample in test
Result of qPCR
Test 1
Test 2
Test 3
Test m
Results after applying Tapestry decoding
47:02 / 1:05:28
David Donoho (Stanford)
SIAM Mathematics of Data Science Plenary Lecture

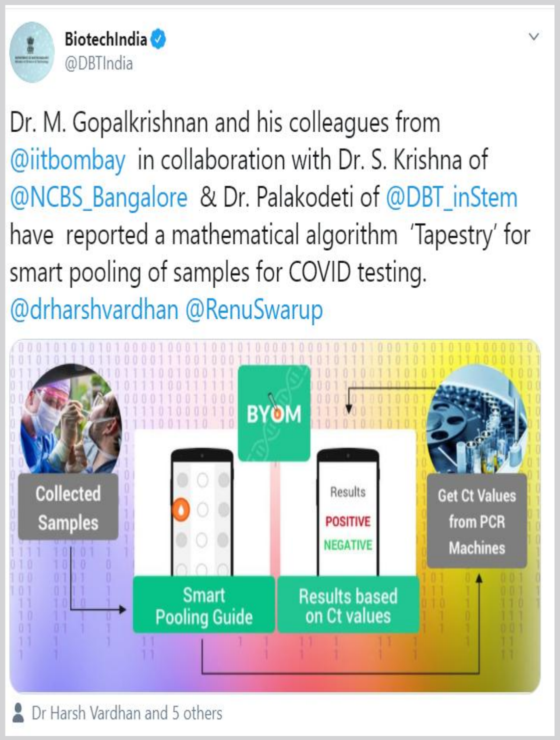


nature

NEWS · 10 JULY 2020

The mathematical strategy that could transform coronavirus testing

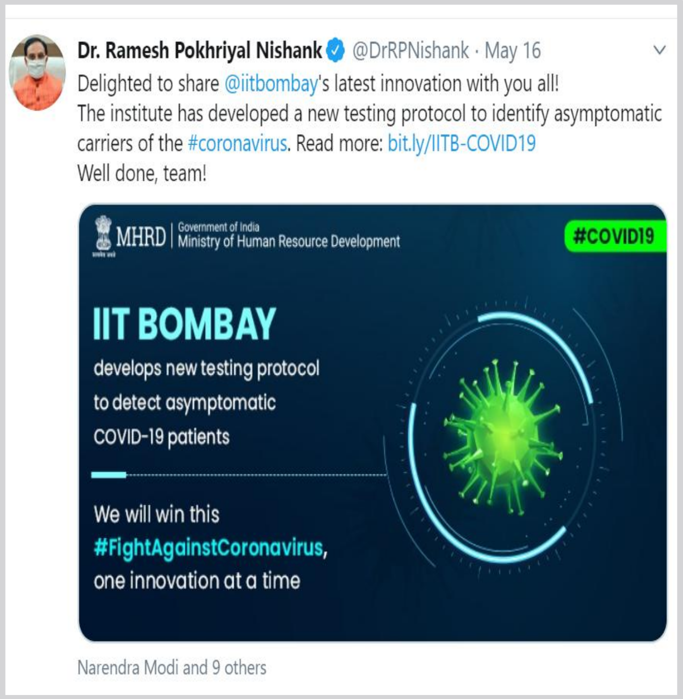
Four charts show how pooling samples from many people can save time or resources.



BiotechIndia @DBTIndia

Dr. M. Gopalkrishnan and his colleagues from @iitbombay in collaboration with Dr. S. Krishna of @NCBS_Bangalore & Dr. Palakodeti of @DBT_inStem have reported a mathematical algorithm 'Tapestry' for smart pooling of samples for COVID testing. @drharshvardhan @RenuSwarup

Dr Harsh Vardhan and 5 others



Dr. Ramesh Pokhriyal Nishank @DrRPNishank · May 16

Delighted to share @iitbombay's latest innovation with you all!
The institute has developed a new testing protocol to identify asymptomatic carriers of the #coronavirus. Read more: bit.ly/IITB-COVID19
Well done, team!

MHRD | Government of India Ministry of Human Resource Development
#COVID19
IIT BOMBAY
develops new testing protocol to detect asymptomatic COVID-19 patients
We will win this #FightAgainstCoronavirus, one innovation at a time

Narendra Modi and 9 others

Summary of Tapestry

- Tapestry is **non-adaptive**, i.e. **single stage** (unlike Dorfman)

- Tapestry has in-built methods to estimate **number of infected people** (*k*) from the RT-PCR test results on pools (resort to COMP if *k* is too high)

- Tapestry handles **noisy** measurements.

- Unlike some binary group-testing methods, Tapestry measures **viral loads**

Summary of Tapestry

- Algorithmic parameters auto-tuned by cross-validation

- Algorithms do not use any prior knowledge of *k*

- In our design of **A**, each sample contributes to exactly 3 pools. Each pool contains contributions from ~40 samples.

Discussion: Practical Aspects

- How many measurements? O(k log n) for random matrices, O(k^2) for deterministic
- Empirically, we see that k log n measurements are good enough for many of our results
- How to predict **minimum** number of **required** measurements?

Discussion: Practical Aspects

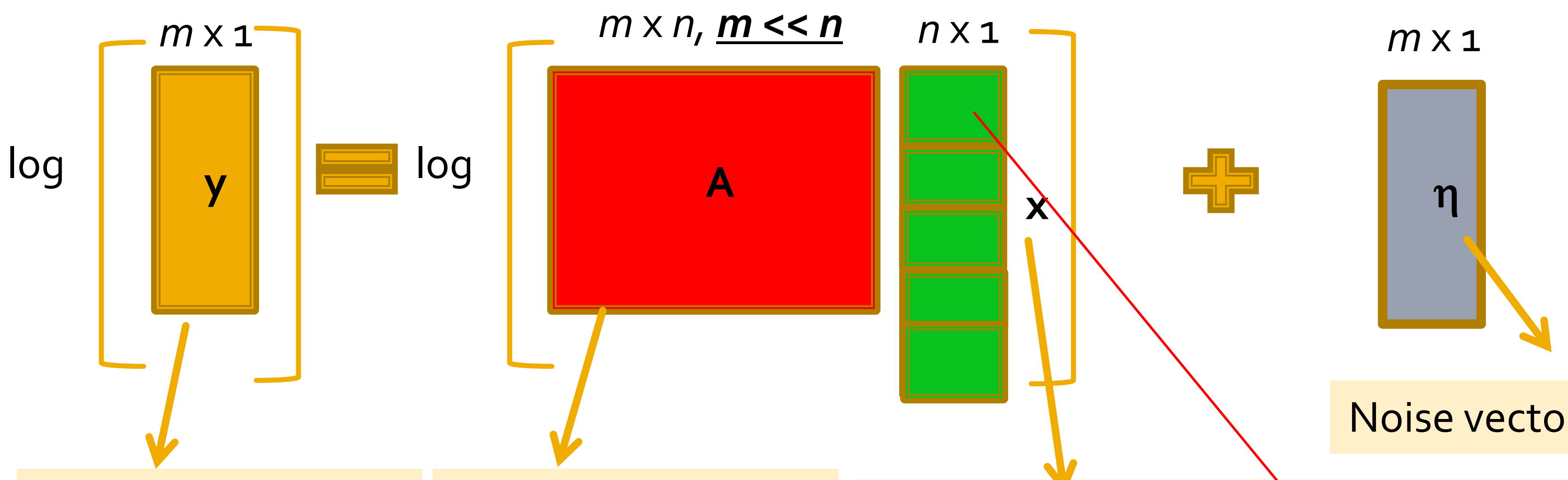
- Can you **guarantee** no false negatives for non-adaptive pooled tests with minimal false positives?
- More accurate noise models?

Discussion: Related Work

- (#) Work from Noam Shental's group (LASSO with Reed-Solomon codes)
- (*) Work from W. Xu's group (synthetic data only)
- ($) Binary PCR work by Dror Baron
- (&) Work by Krishna Narayanan: two stage, adaptive

(#) N. Shental, S. Levy, S. Skorniakov, V. Wuvshet, Y. Shemer-Avni, A. Porgador, and T. Hertz, "*Efficient high throughput SARS-CoV-2 testing to detect asymptomatic carriers*"
(*) J. Yi, R. Mudumbai, and W. Xu, "*Low-cost and high-throughput testing of covid-19 viruses and antibodies via compressed sensing: System concepts and computational experiments*"
($) J. Zhu, K. Rivera, and D. Baron, "*Noisy Pooled PCR for Virus Testing*"
(&) Heiderzadeh and Narayanan, "*Two-stage adaptive pooling with RT-qPCR for COVID-19 screening*"

More prior knowledge: families



Noise vector

Vector of **observed quantities** proportional to viral loads in *m* pools
m << *n*

Known mixing matrix:
A_{ij} = 1 if *j*-th sample contributes to *i*-th pool, and 0 otherwise

Sparse vector of **unknown** viral loads in the samples of *n* different people - **divided into disjoint blocks (eg: families)**

$$J(\mathbf{x}) = \|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1$$

$$J_{group}(\mathbf{x}) = \|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2 + \lambda\sum_i \|\mathbf{x}_{Gi}\|_2$$

Estimate **x** given **y** and **A**

Discussions with Dror Baron.

Group sparsity

$$J(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{x}\|_1$$

$$J_{group}(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda \sum_i \|\boldsymbol{x}_{Gi}\|_2$$

L1 norm is a softened version of the L0 norm – which considers sparsity of **x** [we assumed that the number of infected **individuals** is small]

The group sparsity norm considers the number of infected **families** and assumes it to be small. Counting the number of infected families is called the group-L0 norm but it leads to NP-hard problems (just like the L0 norm). So we soften it to the group-L1 norm. NOTE: **xGi** is a vector that contains the viral load of all members of the i-th family. It is a subvector of **x**.

$$\sum_{i=1}^{F} \|\boldsymbol{x}_{G_i}\|_2 =$$

$$\text{L1 norm of the vector}$$

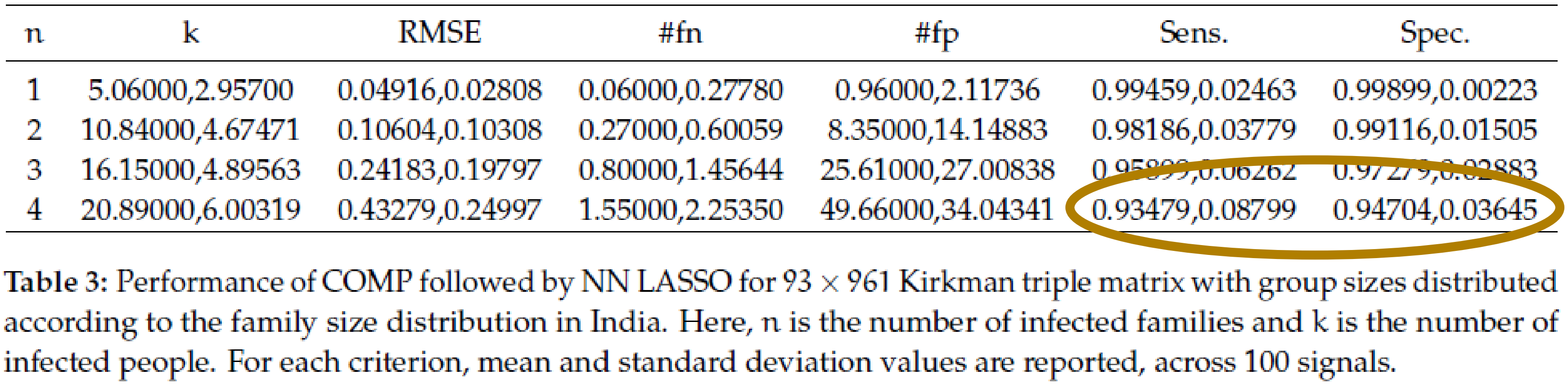
$$(\|\boldsymbol{x}_{G_1}\|_2, \|\boldsymbol{x}_{G_2}\|_2, \ldots, \|\boldsymbol{x}_{G_F}\|_2)$$

$$J(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_1$$

$$J_{group}(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\sum_i\|\boldsymbol{x}_{\boldsymbol{G}i}\|_2$$

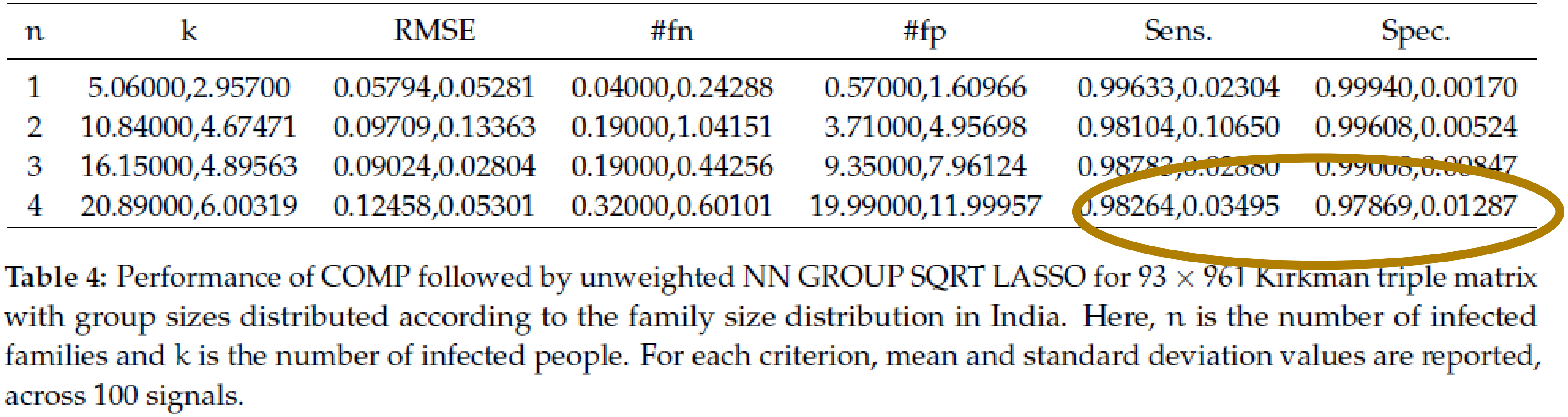
# infected people --> # infected families

Fully Infected Families



n	k	RMSE	#fn	#fp	Sens.	Spec.
1	5.06000,2.95700	0.04916,0.02808	0.06000,0.27780	0.96000,2.11736	0.99459,0.02463	0.99899,0.00223
2	10.84000,4.67471	0.10604,0.10308	0.27000,0.60059	8.35000,14.14883	0.98186,0.03779	0.99116,0.01505
3	16.15000,4.89563	0.24183,0.19797	0.80000,1.45644	25.61000,27.00838	0.95899,0.06262	0.97279,0.02883
4	20.89000,6.00319	0.43279,0.24997	1.55000,2.25350	49.66000,34.04341	0.93479,0.08799	0.94704,0.03645

Table 3: Performance of COMP followed by NN LASSO for 93 × 961 Kirkman triple matrix with group sizes distributed according to the family size distribution in India. Here, n is the number of infected families and k is the number of infected people. For each criterion, mean and standard deviation values are reported, across 100 signals.



n	k	RMSE	#fn	#fp	Sens.	Spec.
1	5.06000,2.95700	0.05794,0.05281	0.04000,0.24288	0.57000,1.60966	0.99633,0.02304	0.99940,0.00170
2	10.84000,4.67471	0.09709,0.13363	0.19000,1.04151	3.71000,4.95698	0.98104,0.10650	0.99608,0.00524
3	16.15000,4.89563	0.09024,0.02804	0.19000,0.44256	9.35000,7.96124	0.98782,0.02880	0.99008,0.00847
4	20.89000,6.00319	0.12458,0.05301	0.32000,0.60101	19.99000,11.99957	0.98264,0.03495	0.97869,0.01287

Table 4: Performance of COMP followed by unweighted NN GROUP SQRT LASSO for 93 × 961 Kirkman triple matrix with group sizes distributed according to the family size distribution in India. Here, n is the number of infected families and k is the number of infected people. For each criterion, mean and standard deviation values are reported, across 100 signals.

Partially Infected Families

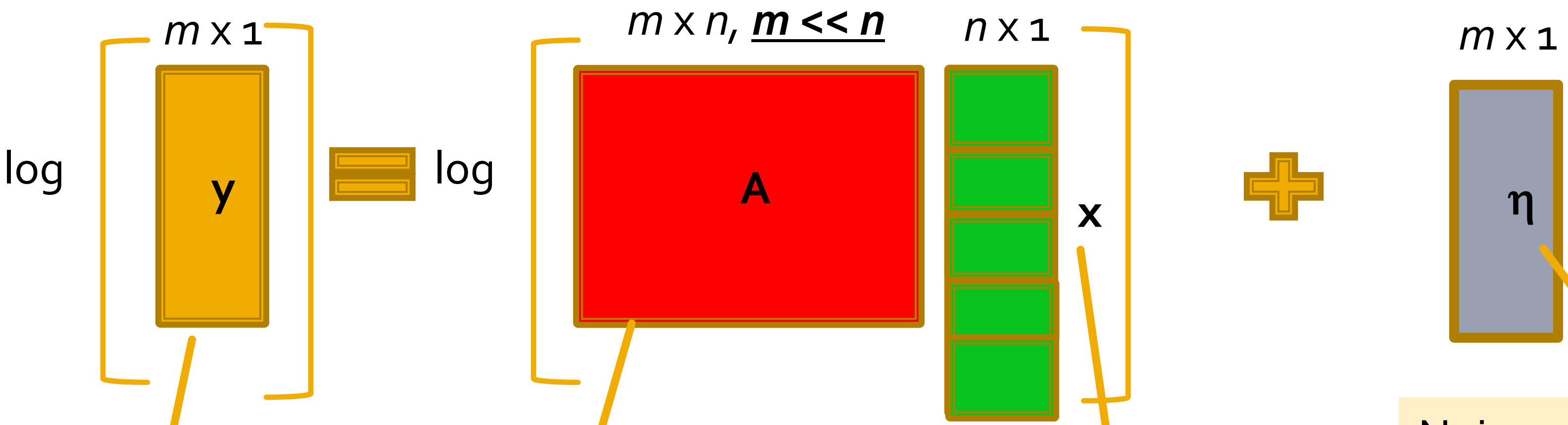
n	k	RMSE	#fn	#fp	Sens.	Spec.
1	5.06000,2.95700	0.04916,0.02808	0.06000,0.27780	0.96000,2.11736	0.99459,0.02463	0.99899,0.00223
2	10.84000,4.67471	0.10604,0.10308	0.27000,0.60059	8.35000,14.14883	0.98186,0.03779	0.99116,0.01505
3	16.15000,4.89563	0.24183,0.19797	0.80000,1.45644	25.61000,27.00838	0.95899,0.06262	0.97279,0.02883
4	20.89000,6.00319	0.43279,0.24997	1.55000,2.25350	49.66000,34.04341	0.93479,0.08799	0.94704,0.03645

Table 7: Performance of COMP followed by NN LASSO for 93 × 961 Kirkman triple matrix with group sizes distributed according to the family size distribution in India. Here, n is the number of infected families, k is the number of infected people and a non-zero number of people between 1 and group size are infected uniformly at random within each family. For each criterion, mean and standard deviation values are reported, across 100 signals.

n	k	RMSE	#fn	#fp	Sens.	Spec.
1	5.06000,2.95700	0.05794,0.05281	0.04000,0.24288	0.57000,1.60966	0.99633,0.02304	0.99940,0.00170
2	10.84000,4.67471	0.09709,0.13363	0.19000,1.04151	3.71000,4.95698	0.98104,0.10650	0.99608,0.00524
3	16.15000,4.89563	0.09024,0.02804	0.19000,0.44256	9.35000,7.96124	0.98783,0.02880	0.99008,0.00847
4	20.89000,6.00319	0.12458,0.05301	0.32000,0.60101	19.99000,11.99957	0.98264,0.03495	0.97869,0.01287

Table 8: Performance of COMP followed by unweighted NN GROUP SQRT LASSO for 93 × 961 Kirkman triple matrix with group sizes distributed according to the family size distribution in India. Here, n is the number of infected families, k is the number of infected people and a non-zero number of people between 1 and group size are infected uniformly at random within each family. For each criterion, mean and standard deviation values are reported, across 100 signals.

Even more prior knowledge: contact tracing



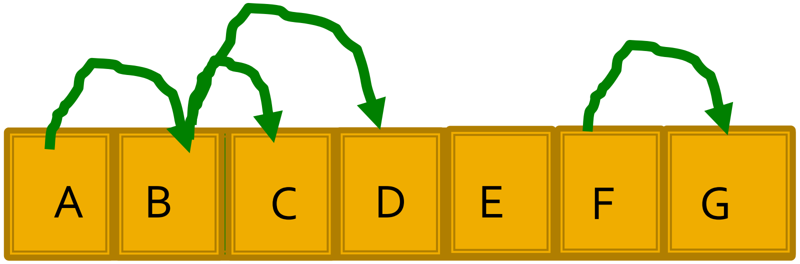
Vector of **observed quantities** proportional to viral loads in *m* pools *m* << *n*

Known mixing matrix: A_{ij} = 1 if *j*-th sample contributes to *i*-th pool, and 0 otherwise

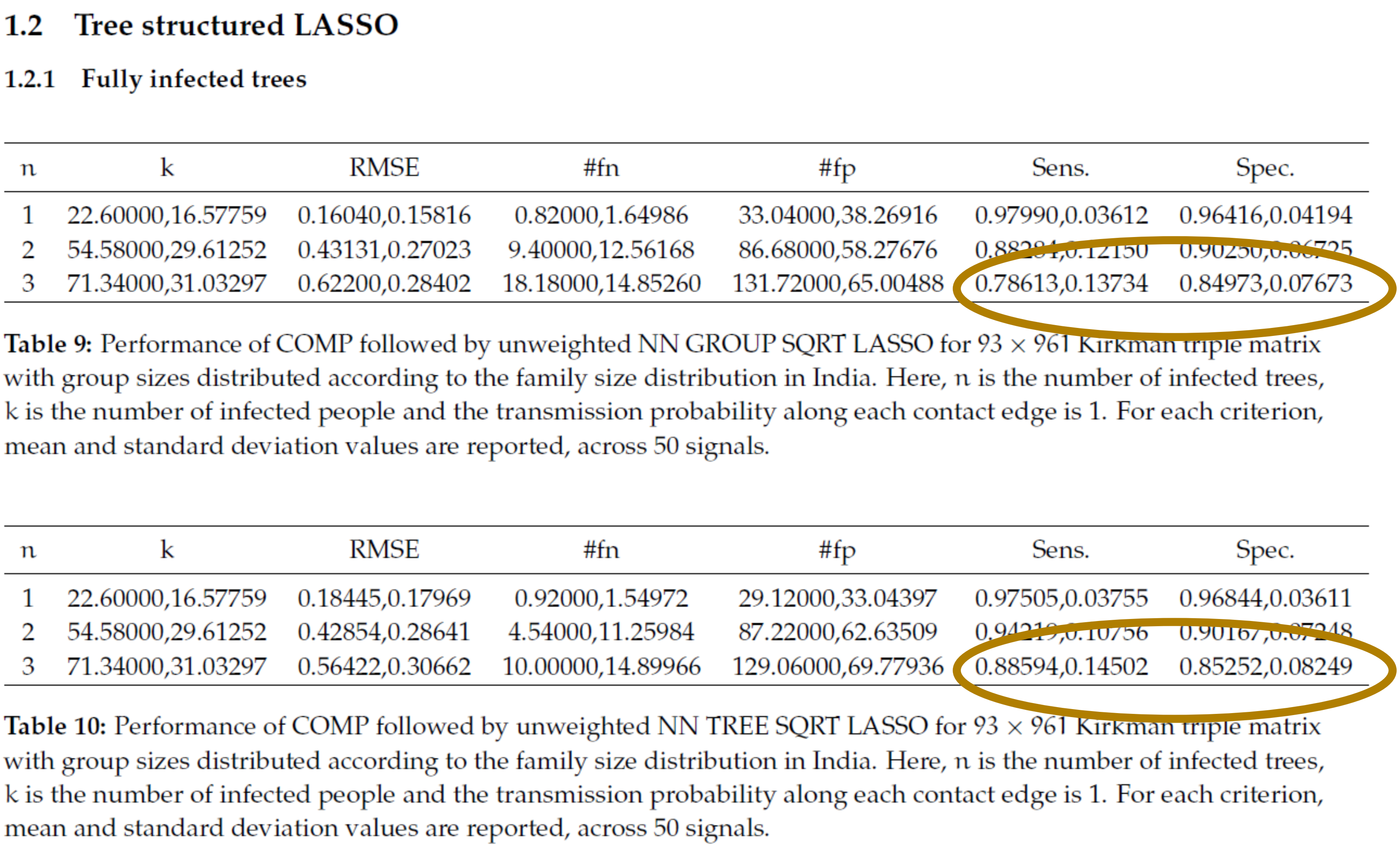
Sparse vector of **unknown** viral loads in the samples of *n* different people - **divided into blocks (eg: families) and you have contact tracing information**

Noise vector

Estimate **x** given **y** and **A**



Even more prior knowledge: contact tracing



1.2 Tree structured LASSO

1.2.1 Fully infected trees

n	k	RMSE	#fn	#fp	Sens.	Spec.
1	22.60000,16.57759	0.16040,0.15816	0.82000,1.64986	33.04000,38.26916	0.97990,0.03612	0.96416,0.04194
2	54.58000,29.61252	0.43131,0.27023	9.40000,12.56168	86.68000,58.27676	0.88284,0.12150	0.90250,0.06725
3	71.34000,31.03297	0.62200,0.28402	18.18000,14.85260	131.72000,65.00488	0.78613,0.13734	0.84973,0.07673

Table 9: Performance of COMP followed by unweighted NN GROUP SQRT LASSO for 93 × 961 Kirkman triple matrix with group sizes distributed according to the family size distribution in India. Here, n is the number of infected trees, k is the number of infected people and the transmission probability along each contact edge is 1. For each criterion, mean and standard deviation values are reported, across 50 signals.

n	k	RMSE	#fn	#fp	Sens.	Spec.
1	22.60000,16.57759	0.18445,0.17969	0.92000,1.54972	29.12000,33.04397	0.97505,0.03755	0.96844,0.03611
2	54.58000,29.61252	0.42854,0.28641	4.54000,11.25984	87.22000,62.63509	0.94219,0.10756	0.90167,0.07248
3	71.34000,31.03297	0.56422,0.30662	10.00000,14.89966	129.06000,69.77936	0.88594,0.14502	0.85252,0.08249

Table 10: Performance of COMP followed by unweighted NN TREE SQRT LASSO for 93 × 961 Kirkman triple matrix with group sizes distributed according to the family size distribution in India. Here, n is the number of infected trees, k is the number of infected people and the transmission probability along each contact edge is 1. For each criterion, mean and standard deviation values are reported, across 50 signals.

More details about the contact tracing work

- Ritesh Goenka, Shu-Jie Cao, Chau-Wai Wong, *Ajit Rajwade*, Dror Baron,"Contact Tracing Enhances the Efficiency of COVID-19 Group Testing", ICASSP 2021 (accepted to Special session on "Data Science Methods for COVID-19"). Arxiv version available.

- Some results on false positive rate (FPR) and false negative rate (FNR) on next slide.

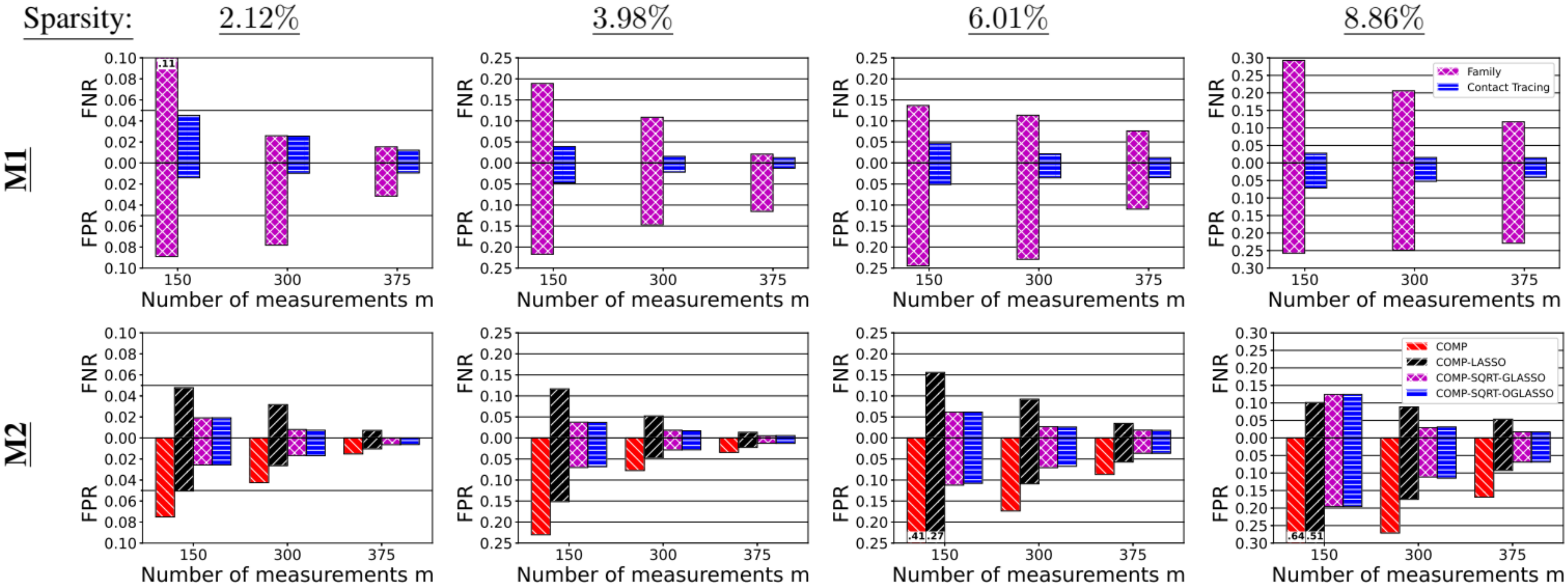


Fig. 3. Performance of the proposed group testing methods **M1** (top row) with binary noise and **M2** (bottom row) with multiplicative noise at four averaged sparsity levels and three measurement levels for a population of $n = 1000$ individuals.

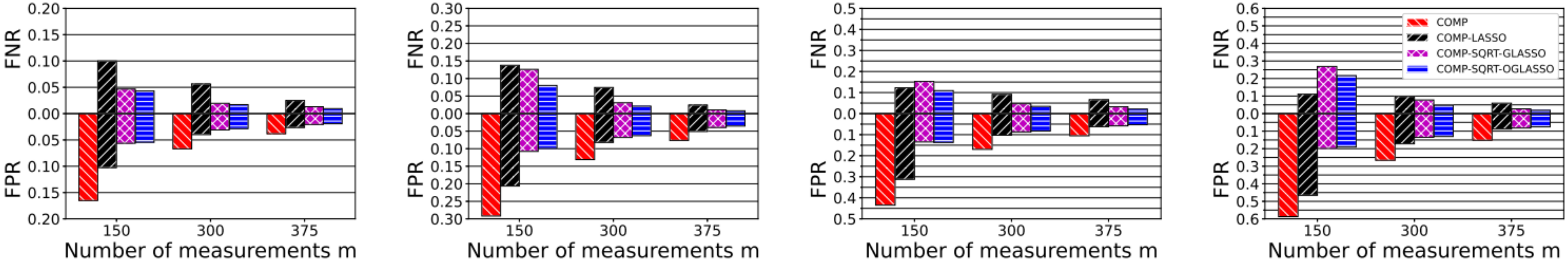


Fig. 6. Figure showing mean FNR and FPR values for the contact graph from Section 4.1, for mean sparsity levels of 3.20%, 4.84%, 6.25%, 8.66% (from left to right).

Thanks for listening!
Questions?
Website:

http://www.cse.iitb.ac.in/~ajitvr